國立臺灣大學電機資訊學院資訊工程學研究所

碩士論文

Department of Computer Science and Information Engineering
College of Electrical Engineering and Computer Science
National Taiwan University

Master's Thesis

張檢測：基於捲積變分自動編碼器快速重建的血糖試片瑕疵檢測

ZhangInspect: Blood Glucose Test Strip Anomaly Detection Based on Fast Reconstruction of Convolutional Variational Auto-Encoder

張子威

Zi-Wei Zhang

指導教授：傅楸善 博士

Advisor: Chiou-Shann Fuh, Ph.D.

中華民國 114 年 7 月

July, 2025

doi:10.6342/NTU202503439

國立臺灣大學碩士學位論文
口試委員會審定書
MASTER'S THESIS ACCEPTANCE CERTIFICATE
NATIONAL TAIWAN UNIVERSITY

張檢測：基於卷積變分自動編碼器快速重建的血糖試片瑕疵檢測

ZhangInspect: Blood Glucose Test Strip Anomaly Detection Based on Fast Reconstruction of Convolutional Variational Auto-Encoder

本論文係張子威君（學號 R11922186）在國立臺灣大學資訊工程學系完成之碩士學位論文，於民國 114 年 07 月 07 日承下列考試委員審查通過及口試及格，特此證明。

The undersigned, appointed by the Department of Computer Science and Information Engineering on 7 July 2025 have examined a Master's thesis entitled above presented by ZHANG, ZIWEI (student ID: R11922186) candidate and hereby certify that it is worthy of acceptance.

口試委員 Oral examination committee:

傅楸善 _____ _____

（指導教授 Advisor）

劉木謙 _____ _____

_____ _____ _____

系主任/所長 Director: 陳祝嵩

doi:10.6342/NTU202503439

誌謝

本論文的完成，首先要感謝傅楸善教授的指導，他總是不遺餘力地在研究過程中提供支持與建議；同時也要感謝晟格公司提供合作機會，支持我研究這個主題；最後也要感謝立寶光電，提供了本論文的所有的樣品，支持了訓練資料集的建立。

另外也要感謝數位相機與電腦視覺實驗室的學長姊和同學在研究期間的照顧。李詠億學長、吳柏緯學長、邱議禾學長、游惟丞學長、郁霈靖學長、凌宇帆學長、方郁婷學姊、張婷淇學姊無論在任何問題總會給予我適當的幫助，讓我在研究過程中順利度過難關。同學陳孝寧、丁浩軒、康家豪、周家弘、鄒雨笙、藍泓景，學妹陳婷，在研究過程中互相扶持照顧，一同分享學業上的甘苦。數位相機與電腦視覺實驗室的所有成員們對於幫助我完成這篇論文實在功不可沒。

最後要感謝我的家人和朋友，父親張銘訓和母親沈卓亞爲我研究期間提供了大部分的生活費，台大的好朋友們如慇語芊、王棪琪、錢泓瑞、張文峰等等總是會在我最需要的時候幫助我，使我能專心投入課業，無後顧之憂。還有僑陸組的以萱老師也不遺餘力在學校方方面面幫助我。在此僅將這段時間的研究成果匯集成本論文，獻給所有曾經關心、照顧與幫助我的所有人。

doi:10.6342/NTU202503439

中文摘要

血糖酶试片是用以检测血糖水平的医学试纸，它在目前的工业生产中虽然以及完成自动化，但是其瑕疵品检测的部分还是需要大量的人力进行目检，人力检测效率和检测率都不稳定，最终会影响工厂的产量和良品率。

所以开发一款自动化的检测设备刻不容缓。但是工业瑕疵检测都面临着瑕疵样本难以获得的问题，而血糖酶试纸瑕疵检测更是面临检测速度要求快，检测精度高，产品类型多样和瑕疵样式不固定等特点，基于这些，我们开发了一款基于VAE[3]的捲積 VAE 和基於 SSM 的損失函數算法的 ZhangInspect 血糖酶試片瑕疵檢測算法，通過變分自動編碼器的圖像重建和去噪功能，巧妙地通過補全瑕疵樣本以找到瑕疵位置，并且通過捲積的方法提取空間結構信息，使得重建結果更加精確。通過以上方法，相較於傳統的重建瑕疵檢測，模型訓練時間和重建時間都大大下降，準確率也非常高，增加了算法的實用性和適用性。

關鍵字：張檢測、血糖酶試片、無監督式學習、瑕疵檢測、影像重建、變分自編碼器、捲積神經網絡

doi:10.6342/NTU202503439

ABSTRACT

Glucose enzyme test strips are medical test strips used to detect blood sugar levels. Although they have been automated in current industrial production, the anomaly detection part still requires much manpower for visual inspection. The efficiency and detection rate of manual inspection are unstable, which will eventually affect the factory's output and yield rate. Therefore, it is urgent to develop an automated inspection device. However, industrial anomaly detection faces the problem of difficulty in obtaining anomaly samples, and glucose enzyme test strip anomaly detection faces the characteristics of fast detection speed, high detection accuracy, diverse product types, and non-fixed anomaly patterns. Based on these, we develop ZhangInspect glucose enzyme test strip anomaly detection algorithm based on the convolution Variational Auto-Encoder (VAE) [3] and the loss function algorithm based on Sum of Square Error (SSE). Through the image reconstruction and denoising function of the variational autoencoder, the anomaly location is found by ingeniously completing the anomaly sample, and the spatial structure information is extracted through the convolution method, making the reconstruction result more accurate. Through the above methods, compared with traditional reconstruction anomaly detection, ZhangInspect training time and reconstruction time are greatly reduced, the accuracy is also very high, and the practicality

doi:10.6342/NTU202503439

and applicability of the algorithm are increased.

Keywords: ZhangInspect, glucose enzyme test strip, unsupervised learning, anomaly detection, image reconstruction, variational autoencoder, convolutional neural network

doi:10.6342/NTU202503439

目　次

doi:10.6342/NTU202503439

doi:10.6342/NTU202503439

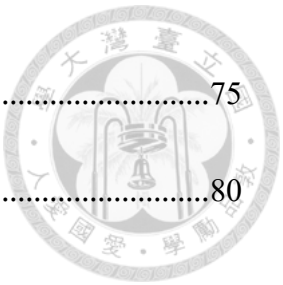
doi:10.6342/NTU202503439

圖　次

doi:10.6342/NTU202503439

doi:10.6342/NTU202503439

doi:10.6342/NTU202503439



doi:10.6342/NTU202503439

表　次

doi:10.6342/NTU202503439

Chapter 1 Introduction

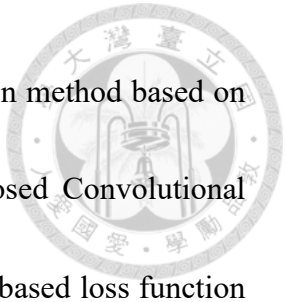
1.1 Overview

Industrial product anomaly detection on modern production lines is different from other computer vision technologies that target detection, and will face some unique problems. For example, the pursuit of efficiency in factories must require a certain detection speed; the detection accuracy will also affect the reputation and sales of the product; the rapid changes in the market will cause rapid product updates, and anomaly detection equipment must be able to quickly switch detection targets. For this reason, we have listed 3 requirements for anomaly detection on industrial product lines:

- Fast detection speed,
- High detection accuracy,
- Fast product batch updates.

With the advancement of technology, machine learning computer vision is increasingly used in industrial product anomaly detection. However, the number of anomaly samples for industrial product anomaly detection is usually very small, and unpredictable and unclassifiable anomalies may occur during the production process. This makes supervised learning ineffective in the field of industrial product anomaly detection. Correspondingly, unsupervised learning is becoming increasingly important in the field of industrial product anomaly detection, because it mainly requires normal

doi:10.6342/NTU202503439



samples. Our ZhangInspect is an industrial product anomaly detection method based on unsupervised learning image reconstruction. It combines our proposed Convolutional Variational Auto-Encoder (CVAE) and Sum of Squared Error (SSE)-based loss function calculation method, making the model's reconstruction efficiency 119 times faster than that of conditioned Denoising Diffusion Models for Anomaly Detection (DDAD) [1] reconstruction models, and the model's training efficiency is more than 523 times faster than that of DDAD reconstruction models. The model's detection accuracy is also improved by 18.5%.

1.2 Introduction of Blood Glucose Enzyme Test Strip

The blood glucose enzyme test strip is our main object. Its composition from bottom to top is PET (Poly-Ethylene Terephthalate), conductive silver ink, conductive carbon ink, and plastic protective film [13] in Figure 1-2-1. The working principle of the blood glucose enzyme test strip is that there is a display area and a reaction area on the surface of the test strip. The enzyme in the reaction area will react with the blood to generate gluconic acid and potassium ferrocyanide. The blood glucose meter will apply a constant voltage to the test strip to oxidize potassium ferrocyanide into potassium ferrocyanide, generating an oxidation current. The magnitude of the oxidation current is proportional

doi:10.6342/NTU202503439

to the blood glucose concentration. The blood glucose meter records the magnitude of the oxidation current and converts it into blood glucose concentration.

Factories will produce various types of blood glucose test strips. Figure 1-2-2 is a blood glucose test strip with piece type Connected, and Figure 1-2-3 is a blood glucose test strip with piece type Not Connected. The same piece type can have multiple different piece IDs (Identity). The difference between different piece IDs is that the small pieces in the large piece are different in shape in Figure 1-2-4. Therefore, we need to first develop an adaptive cutting algorithm to cut the large test strip into small pieces, and then detect whether the small pieces have anomalies. Thus, we develop a preliminary Automated Optical Inspection (AOI) program to cut and identify anomalies on glucose enzyme test strips. However, due to the rapid product updates, the AOI algorithm update requires much manpower, and the detection speed is high, so a machine learning-based detection model is still needed for detection.

doi:10.6342/NTU202503439

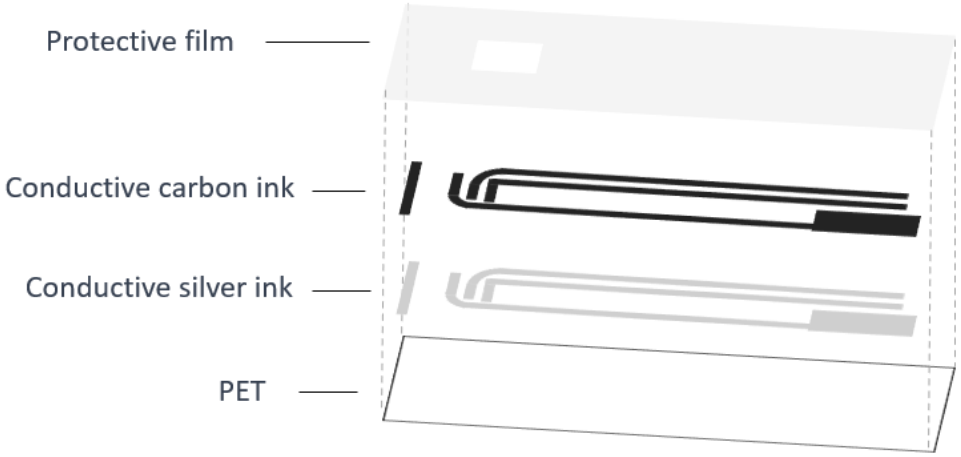


Figure 1-2-1: Architecture of blood glucose enzyme test strip. [13] composed of PET (Poly-Ethylene Terephthalate), conductive silver ink, conductive carbon ink, and protective film.

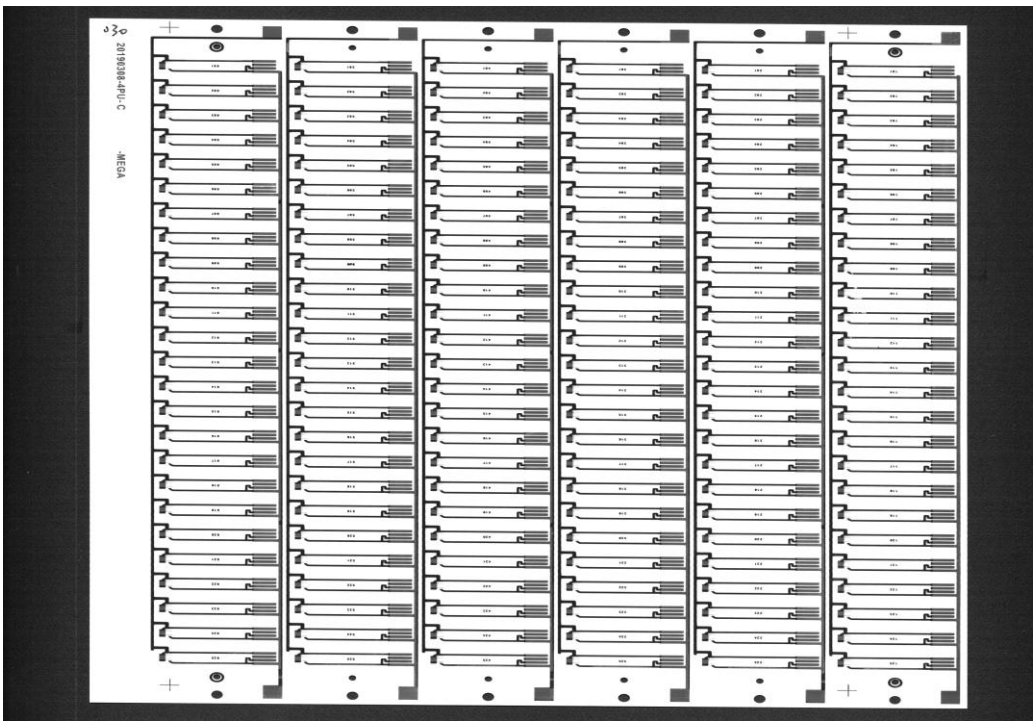


Figure 1-2-2: Blood glucose enzyme test strip. piece type: Connected, piece ID: UA.

doi:10.6342/NTU202503439

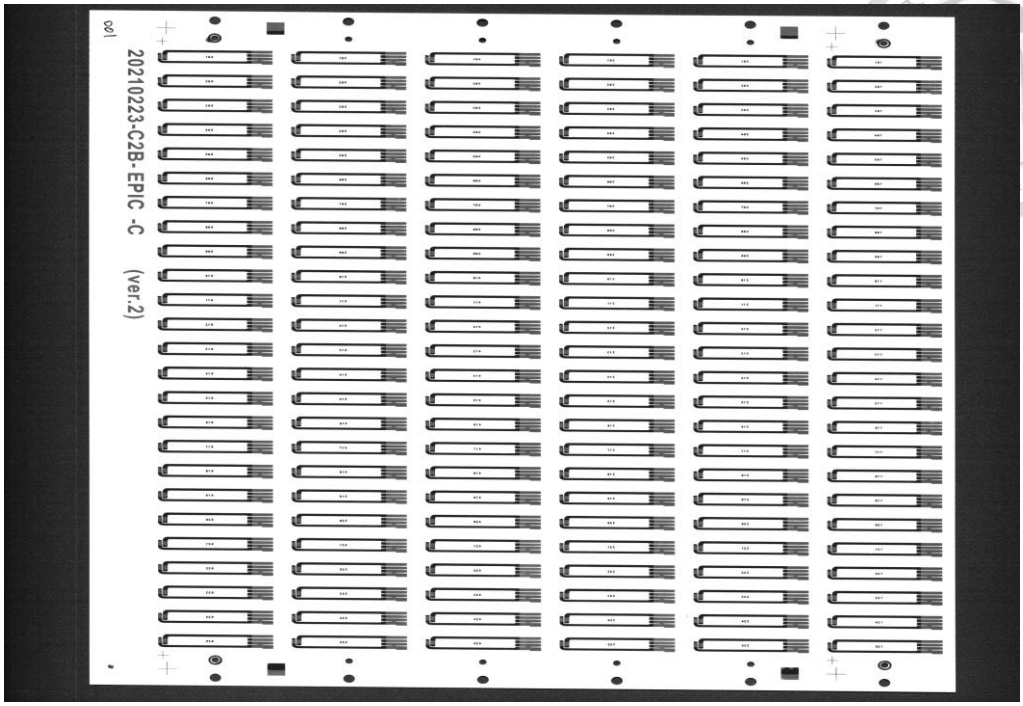


Figure 1-2-3: Blood glucose enzyme test strip. piece type: Not Connected, piece ID: C2B.

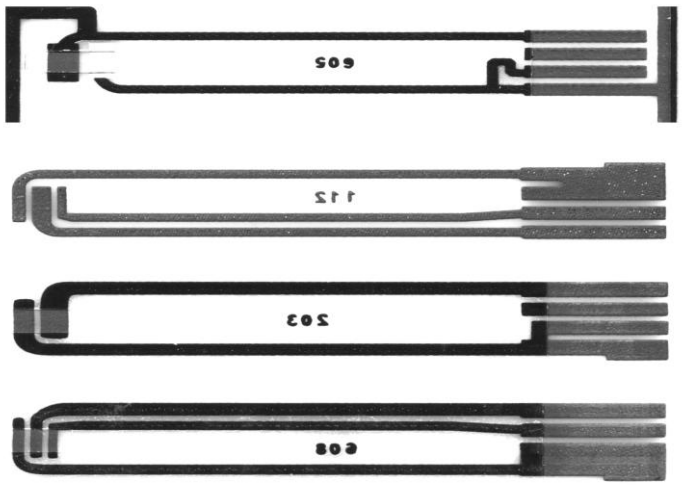


Figure 1-2-4: Different piece IDs of blood glucose enzyme test strips. From top to bottom, piece IDs: UA, V5D, C2B, M4A2.

doi:10.6342/NTU202503439

1.3 Inspection Environment

The blood glucose enzyme test strip anomaly detection machine adopts an assembly line design. To protect the test strip, we choose to use a suction cup to pick up the test strip and place it on the conveyor belt to send it to the detection area (Figure 1-3-1). The detection area uses a line light source CMOS (Complementary Metal Oxide Semiconductor) and a 16K line scan camera (Dalsa p3-80-16k40) to collect the image of the blood glucose enzyme test strip. Of course, due to the vibration of the conveyor belt and the unevenness of the forward movement, the image quality will be reduced, which will affect the image detection effect to a certain extent.

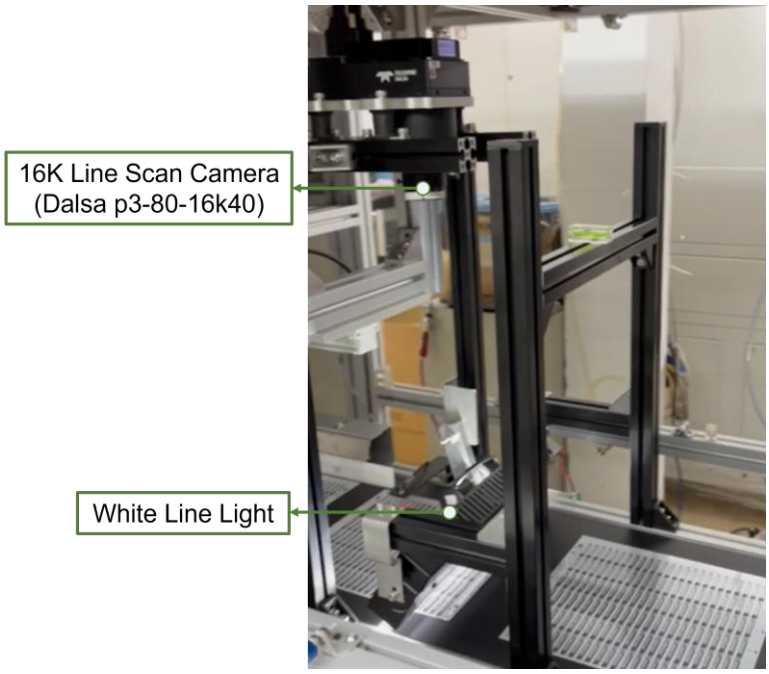


Figure 1-3-1: Blood glucose enzyme test strip anomaly detection production line machine.

doi:10.6342/NTU202503439

1.4 Blood Glucose Enzyme Test Strip AOI Detection

Figure 1-4-1 shows the detection process Z-AOI of the AOI algorithm in ZhangInspect. After we use the line scan camera of the machine (Figure 1-3-1) to obtain the line scan image, we use the image merge algorithm to stitch it into a complete test piece image (size is about 15,000*18,000 pixels). Then use the small piece cutting algorithm to segment individual small pieces (depending on the size of the large piece, a large piece can be divided into 150-300 small pieces). For AOI anomaly detection, the detection logic of the anomaly position distribution of the glucose enzyme test piece on the toner (black area) and the white background (white area) is different, so it is necessary to segment the toner and white background. Finally, we need to perform anomaly detection on the two areas separately to output the anomaly detection results and the dataset required for subsequent training. Because there is no standardized anomaly regulation in the actual factory, we make a simple classification of anomalies in Figure 1-4-2, including blue pen lines: blue pen markings on the test piece by the manufacturer; lack of ink: there are large missing or scratches in the toner area; ink halo and foreign matter: there is toner or impurities remaining in the white area.

doi:10.6342/NTU202503439

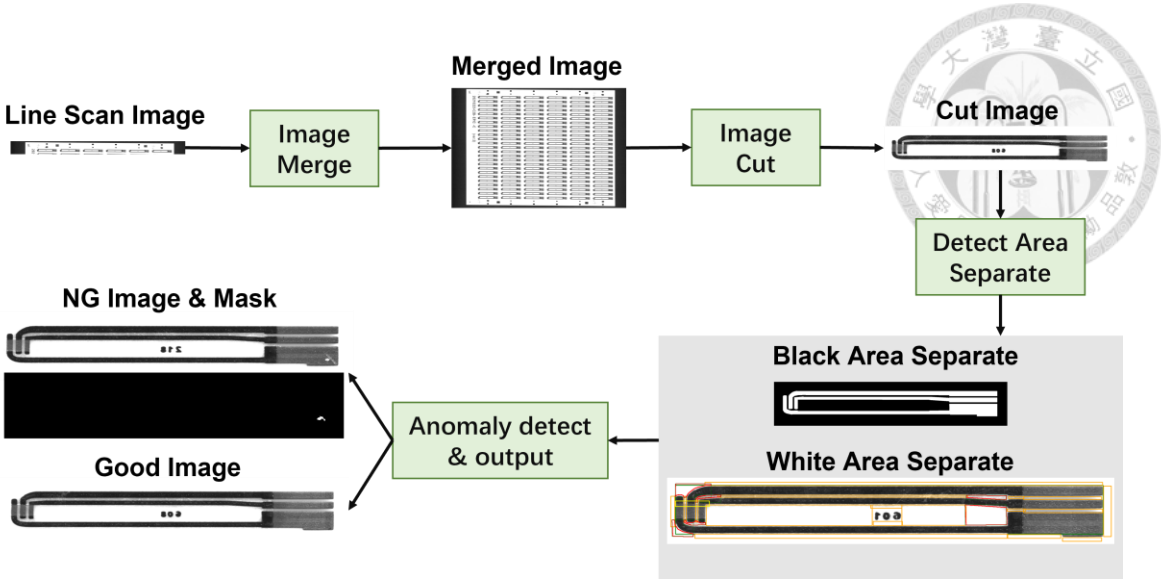


Figure 1-4-1: Blood glucose enzyme test piece AOI anomaly detection and dataset output process (Z-AOI).

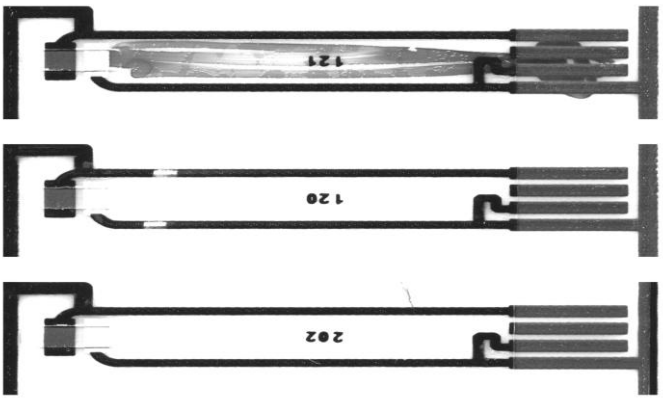


Figure 1-4-2: The types of anomalies on the blood glucose enzyme test strips, from top to bottom, are: blue pen lines, missing ink, and blurred ink.

doi:10.6342/NTU202503439

1.5 Thesis Organization

This paper will introduce some unsupervised anomaly detection models in Chapter 2. Chapter 3 will introduce some techniques we use and why we use them. Chapter 4 will introduce our ZhangInspect and the originality of the technology it developed. Chapter 5 will show and discuss the experimental results of ZhangInspect. Chapter 6 will give a summary and future prospect.

doi:10.6342/NTU202503439

Chapter 2 Related Works

2.1 Overview

This section will introduce some unsupervised learning methods for industrial product anomaly detection and discuss their advantages and disadvantages. We have collected and organized these methods, but there are some limitations in solving the problem of defect detection of blood glucose enzyme test strips. However, they are still very valuable reference methods. We will classify these methods into four categories: reconstruction-based, representation-based, generation-based, and hybrid methods.

2.2 Unsupervised Deep Learning Anomaly Detection

Anomaly detection aims to identify patterns or points that are significantly different from the majority of data, called anomalies or outliers. In the unsupervised setting, it is assumed that the training data mainly consists of normal samples, and the abnormal samples are rare and unlabeled. Unsupervised deep learning anomaly detection uses the powerful representation ability of deep neural networks to detect anomalies by learning complex features of normal data. This method has wide applications in industrial quality control, medical image analysis, network security, and other fields. It is particularly

doi:10.6342/NTU202503439

suitable for scenarios where anomalies are rare because it does not require labeled abnormal data.

Deep learning provides various techniques for unsupervised anomaly detection, including Auto-Encoders (AE) [2] (Figure 2-2-1), Variational Auto-Encoders (VAEs) [3], Generative Adversarial Networks (GANs) [4], single-class classification methods, and the recent diffusion model. These methods achieve anomaly detection through different modeling strategies such as reconstruction, representation learning, or probability distribution modeling. The following is a brief introduction to these methods:

Autoencoder: An autoencoder is a neural network that contains an encoder and a decoder. The training goal is to compress the input data into a low-dimensional representation and then reconstruct it into the original data. Since they are trained only on normal data, autoencoders have poor reconstruction effects on abnormal data, and anomalies can be detected through reconstruction errors.

Variational Auto-Encoders (VAEs) [3]: VAEs introduce probabilistic modeling based on autoencoders to learn the potential distribution of data. Anomalies usually have a low generation probability or a high reconstruction error.

Generative Adversarial Networks (GANs) [4]: GANs consist of a generator and a discriminator. The generator learns to generate normal data, and the discriminator

doi:10.6342/NTU202503439

distinguishes between real data and generated data. Anomalies can be detected through the output of the discriminator or the reconstruction ability of the generator.

Single-class classification: Such as Support Vector Data Description (SVDD) [6] and the more advanced Deep Support Vector Data Description (Deep SVDD) [5], the training model maps normal data to a compact area (hypersphere) in the feature space, and anomalies are located outside this area. This method can also be combined with VAE, but it is good at classifying anomalies. The current algorithm does not require too many classification functions, and the computing speed is still the most critical part.

Diffusion model [1]: Diffusion model is an emerging generative model that models data distribution through a stepwise denoising process. It has recently been used for anomaly detection, providing high-quality reconstruction and accurate anomaly localization. This method is our main comparison object in this article.

These methods have their own advantages and are suitable for different data types and application scenarios. For example, autoencoders and VAEs are suitable for images and time series data, but single-class classification is suitable for high-dimensional feature spaces, and diffusion models perform well in visual anomaly detection. In general, we divide these methods into four categories, and we will give some actual models in each category to explain their structure and effects in detail.

doi:10.6342/NTU202503439

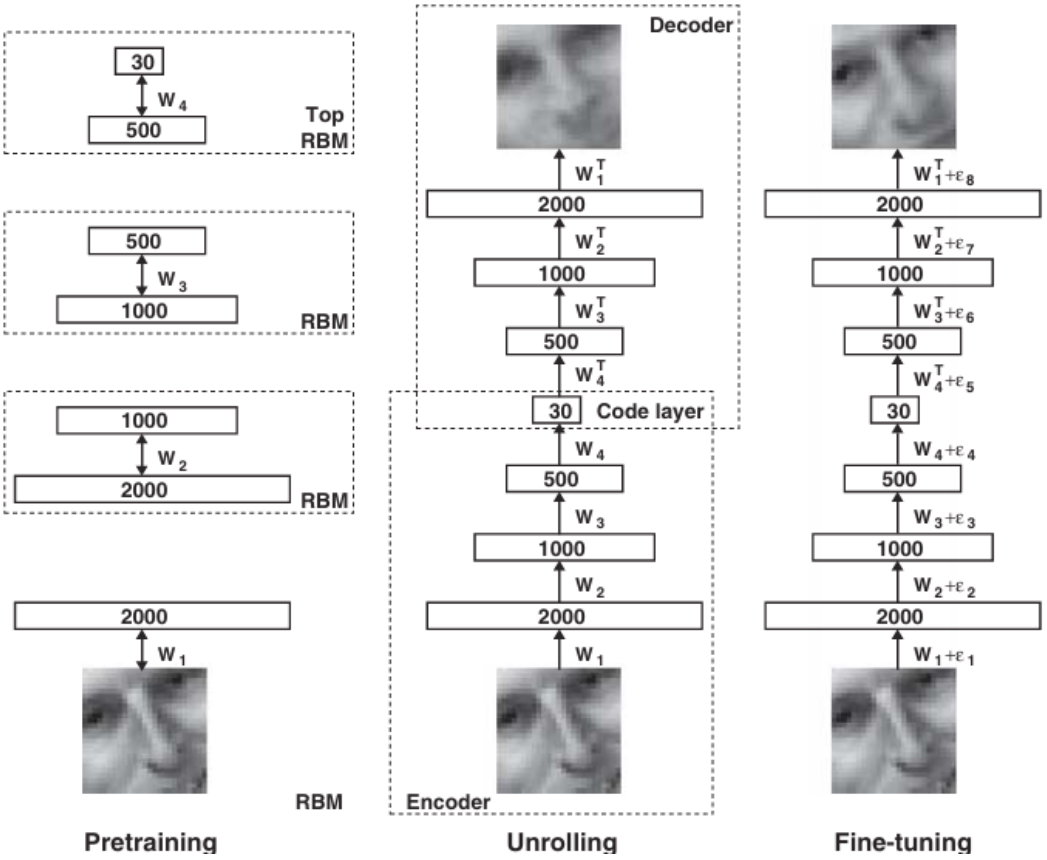


Figure 2-2-1: Pretraining consists of learning a stack of Restricted Boltzmann Machines (RBMs) [2]. each having only one layer of feature detectors. The learned feature activations of one RBM are used as the ‘‘data’’ for training the next RBM in the stack. After the pretraining, the RBMs are ‘‘unrolled’’ to create a deep autoencoder, which is then fine-tuned using backpropagation of error derivatives.

doi:10.6342/NTU202503439

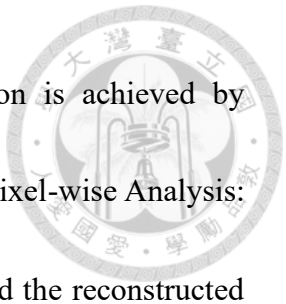
2.2.1 Reconstruction-Based Method

Reconstruction-based methods reconstruct normal data by training models. It is assumed that abnormal data is difficult to be accurately reconstructed, because it has not been seen in training. Anomalies are detected by high reconstruction errors or pixel/feature differences.

Among them, Denoising Diffusion Anomaly Detection (DDAD) [1] is a good reconstruction method for anomaly detection. DDAD is a reconstruction method based on a conditional denoising diffusion model (Figure 2-2-1-1), which generates anomaly-free reconstructed images by guiding the denoising process. During training, the model only uses normal (anomaly-free) samples to learn how to reconstruct normal images from noise. During inference, an image that may contain anomalies is input, and the model generates an anomaly-free reconstructed image through a conditional mechanism, and detects anomalies by comparing the difference between the input and the reconstructed image.

The core of the conditional mechanism is to use the input image as the target, guiding the denoising process to retain normal patterns and remove anomalies. In addition, the paper also proposes an unsupervised domain adaptation technique to fine-tune the feature extractor by generating samples similar to the target image (usually normal samples) to

doi:10.6342/NTU202503439



enhance the effect of feature-level comparison. Anomaly detection is achieved by comparing the input image with its reconstructed image, including Pixel-wise Analysis: directly comparing the pixel differences between the input image and the reconstructed image, calculating the reconstruction error at the pixel level, and Feature-wise Analysis: using a pre-trained feature extractor (such as a feature extractor based on a deep convolutional network) to extract image features and compare the feature differences between the input and the reconstruction. To enhance the effect of feature-level comparison, the model also has an unsupervised domain adaptation technique that fine-tunes the pre-trained feature extractor by generating samples similar to the target image. At the same time, distillation loss is introduced to prevent the fine-tuning process from losing the generalization ability of the pre-trained model, thereby maintaining the robustness of the feature extractor.

Its denoising diffusion model consists of two parts: the forward process and the reverse process. The forward process is to gradually add Gaussian noise to the original image. The reverse process starts with an image containing Gaussian noise and continuously denoises it to make the image gradually clearer until a clean and clear image is generated.

However, this process requires repeated transmission of the forward process and the

doi:10.6342/NTU202503439

reverse process, usually more than 10,000 times, which consumes a lot of computing resources, so the reconstruction efficiency is poor and it is not suitable for real-time scenarios. Secondly, its denoising process is based on the input image. If the anomaly is not very discriminative, the anomaly will be completely reconstructed and cannot be detected, so it has limitations in the anomaly detection of blood glucose enzyme test strips.

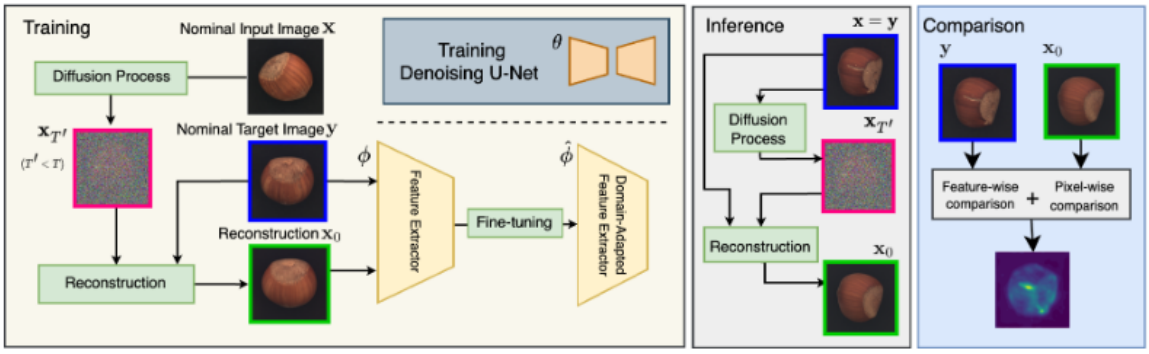


Figure 2-2-1-1: DDAD architecture [1].

2.2.2 Representation-Based Method

Representation learning-based methods detect anomalies by learning the feature representation of normal data because their representation is inconsistent with normal data. Common strategies include single-class classification and teacher-student model. The most common single-classification method is the SVDD [6] method.

The core idea of SVDD is to map the original data to a higher-dimensional feature space through the kernel trick, and then find a hypersphere with the smallest volume in

doi:10.6342/NTU202503439

this space to include most of the normal data points [7] (Figure 2-2-2-1). After the model training is completed, any data point that falls outside the hypersphere will be considered anomaly. The goal of this model is to minimize the volume of the hypersphere while allowing some data points (outliers) to exist outside the sphere, and a hyperparameter is used to make a trade-off.

As a classic anomaly detection algorithm, SVDD has been proven to be effective in many applications, especially in the fields of computer network intrusion detection and industrial fault diagnosis. Its main advantage lies in its unsupervised learning characteristics, which only requires normal samples for training. This is very practical in real industrial scenarios, because in these scenarios, most of the collected data are normal, while fault data are very scarce.

However, the performance of SVDD is largely limited and affected by the selected kernel function. Choosing a suitable kernel function for a specific dataset usually requires expertise and experiments. Moreover, SVDD is a "shallow learning" model. When faced with nonlinear data with complex intrinsic structures, it may not be able to fully learn data features, resulting in poor detection performance [8].

Therefore, Deep SVDD is an extension of traditional SVDD, which combines the powerful feature learning ability of deep learning with the single classification idea of

doi:10.6342/NTU202503439

SVDD. Deep SVDD uses deep neural networks (rather than fixed kernel functions) to learn the mapping from input space to output feature space [5]. Its goal is to train this network so that the feature representation of normal data samples is mapped to a preset compact hypersphere in the feature space, while the feature representation of abnormal samples is mapped to a position far from the center of the sphere. By minimizing the average distance from the network output feature to the center of the sphere, the network is forced to learn the common features and main change patterns in the data. Compared with SVDD, it has more advantages when dealing with complex data with complex hierarchical structures. With the help of deep neural networks, the model can automatically learn complex patterns and discriminative features in the data, avoiding the tedious and expertise-required feature engineering steps in traditional methods.

However, as a deep learning model, the performance of Deep SVDD depends on a large amount of training data. And since the main goal of the model is to compress normal features into a compact sphere, there may be two situations for feature points in the hypersphere. One is that the feature points are concentrated on the surface of the sphere, especially in some datasets where the defects are not obvious. Because the normal and abnormal data are too close, it may not be able to accurately capture the subtle differences between normal data and abnormal data, and thus it is insensitive to some subtle

doi:10.6342/NTU202503439

anomalies. Many of our samples are in this situation. The second is that the feature points are concentrated around the center of the hypersphere, also known as hypersphere collapse. When training the network to map the features of all normal samples to the center *c* of the hypersphere, if there is no constraint, the network may learn a "shortcut" - it maps any input (whether normal or abnormal) to the same point *c*. In this way, the loss of SVDD (the distance to the center) becomes extremely small, but the model completely loses its ability to distinguish and becomes useless.

Finally, the performance of the model is easily affected by the choice of network structure and weight initialization. In the unsupervised case, it is quite difficult to select the optimal model structure and initialization parameters.

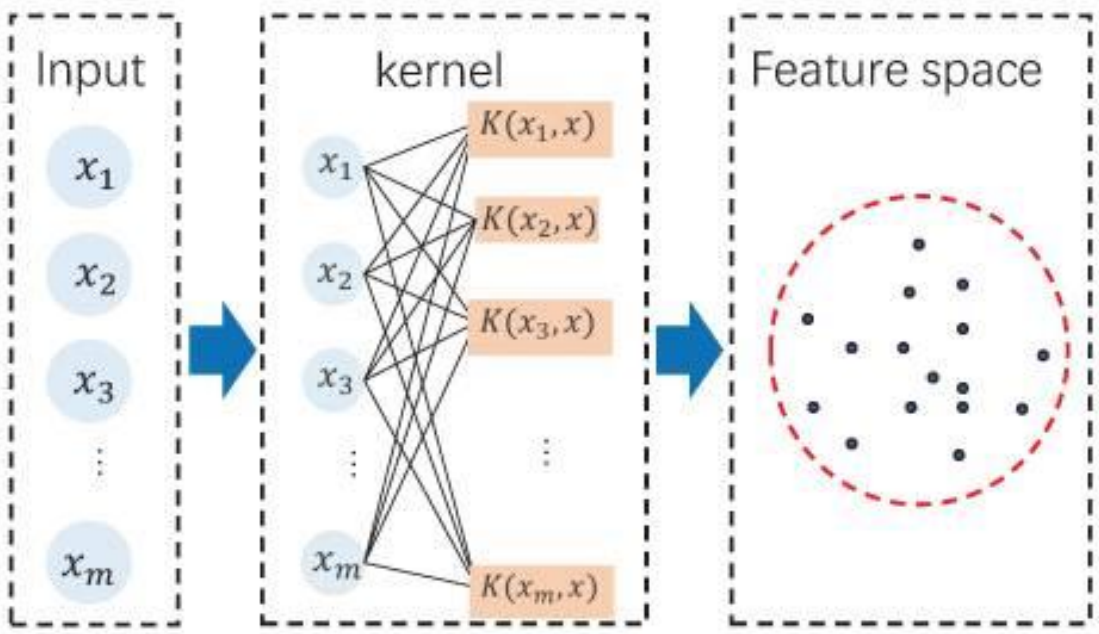


Figure 2-2-2-1: Support Vector Data Description (SVDD) schematic [7].

doi:10.6342/NTU202503439

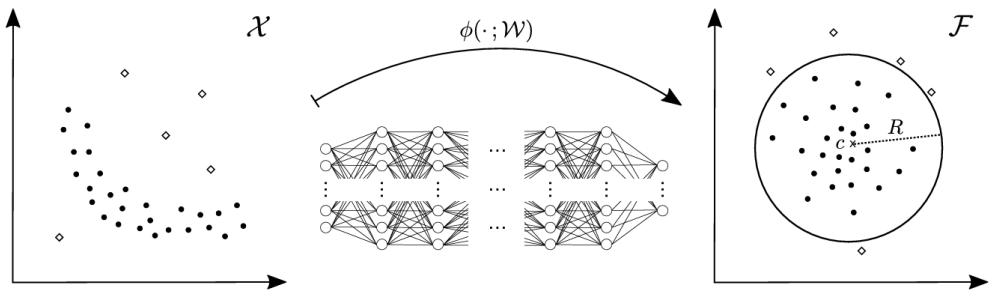


Figure 2-2-2-2: Deep SVDD classification. [5] Deep SVDD learns a neural network transformation $\emptyset(\cdot;\mathcal{W})$ with weights $\mathcal{W}$ from input space $\mathcal{X} \subseteq \mathbb{R}^{\mathrm{d}}$ to output space $\mathcal{F} \subseteq \mathbb{R}^{\mathrm{p}}$ that attempts to map most of the data network representations into a hypersphere characterized by center c and radius R of minimum volume. Mappings of normal examples fall within, whereas mappings of anomalies fall outside the hypersphere.

2.2.3 Generation-Based Method

Generative model-based methods detect anomalies by modeling the probability distribution of normal data, because their probability under this distribution is low. GANs perform well in image anomaly detection, but training instability is a challenge. Among them, Anomaly Detection with Generative Adversarial Networks (AnoGAN) [9] (Figure 2-2-3-1) is a groundbreaking and often cited typical example.

The core idea of AnoGAN is that if a GAN can only learn and generate normal

doi:10.6342/NTU202503439

images, then when it faces an image with an anomaly, it will not be able to perfectly reproduce the anomaly. The method is mainly divided into two stages:

Training stage: First, a large number of anomaly-free product images are collected as training sets. These normal images are used to train a standard GAN. In this process, the Generator learns to generate a normal image that is indistinguishable from a random latent vector, while the Discriminator learns to distinguish between real normal images and fake images generated by the generator.

Detection stage: When a new test image needs to be detected, AnoGAN does not directly input it into the network.

Instead, it freezes the weights of the already trained generator and discriminator. Then, it repeatedly searches in the latent space to find an optimal latent vector z so that the image $G(z)$ generated by this vector through the generator is most similar to the input test image x. An "Anomaly Score" is defined by calculating the difference between the test image x and the reconstructed best normal image $G(z)$. If the test image itself is normal, then the generator can reconstruct it well and the anomaly score is low. If the test image contains anomalies, since the generator has never seen anomalies, it will generate the closest but anomaly-free image, resulting in a large residual and a high anomaly score. By setting a threshold, when the anomaly score exceeds the threshold, the image is judged

doi:10.6342/NTU202503439

as a defective product. At the same time, this residual image itself can be used as an anomaly localization map to clearly mark the location of the anomaly.

With the powerful ability of deep networks, GAN can learn and simulate very complex data distributions (such as cloth textures, metal surfaces, and so on), and has a good effect on anomaly detection in complex backgrounds that are difficult to handle by traditional methods. We can find that the AnoGAN model uses t-SNE (the main goal of t-SNE is to project a high-dimensional data set into a low-dimensional space while preserving the local proximity relationship between data points as much as possible.) After representing it, we can see that there is a good classification of metal surface anomaly.

However, in the detection phase, an iterative optimization is required for each new test image to find the best latent vector *z*, which is very time-consuming and difficult to apply to industrial production lines that require high-speed, real-time detection. Secondly, each small test strip of our blood glucose enzyme test strip has a number in the middle. In the same large test strip, these numbers are different but very similar, which makes it very difficult to find the best z. Generating an image with the wrong number will lead to detection failure.

doi:10.6342/NTU202503439

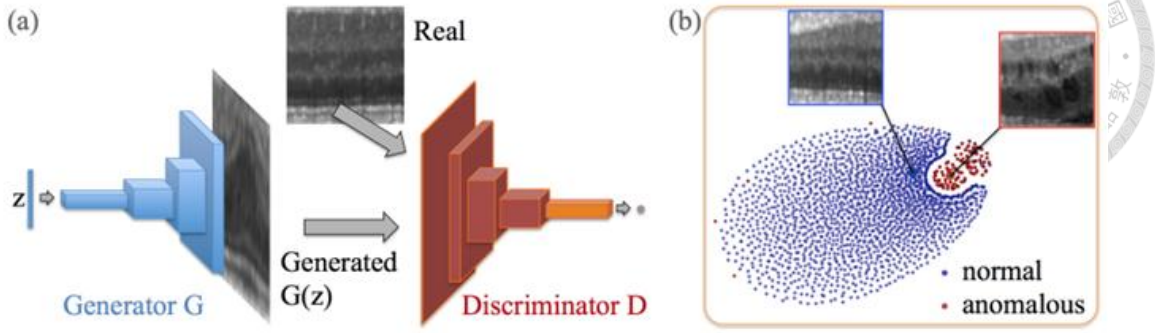


Figure 2-2-3-1: AnoGAN network and t-distributed Stochastic Neighbor Embedding (t-SNE). [9] (a) Deep convolutional generative adversarial network. (b) t-SNE embedding of normal (blue) and anomalous (red) images on the feature representation of the last convolution layer (orange in (a)) of the discriminator.

2.2.4 Hybrid Method

Hybrid methods combine multiple strategies to improve detection performance, especially in complex scenarios. Here we introduce the Accurate Visual Anomaly Detection at Millisecond-Level Latencies (EfficientAD) [10] and VAE-based Deep SVDD [11] methods.

EfficientAD uses a teacher-student model (Figure 2-2-4-1). First, a deep neural network pre-trained on ImageNet, namely the backbone network (Pre-trained Backbone), is used to extract image features, and two independent and specialized teacher models are designed to capture the features of normal data from different scales. The Local Teacher

doi:10.6342/NTU202503439

model focuses on the local details and texture of the image. It divides the image into small patches, and then processes the patches of the image through an AE (called Patch Description Network in their paper). It can upgrade a 33*33*3 image to 1*1*384, thereby reducing the amount of calculation. The Local Student model learns how to reconstruct the features of these patches by imitating the 384-dimensional output of the Local Teacher model. The Global Teacher model focuses on the structure and context of the entire image. It also uses a structure similar to AE, but processes the features of the entire image and learns global representations. The Student Model is a very lightweight fully convolutional network. The input is also the feature map from the pre-trained backbone network. The goal is to imitate the 384-dimensional output of the above two teacher models and obtain the image through the decoder of AE.

The advantage of EfficientAD is that it is extremely efficient. Since only lightweight student models are used during detection, its inference speed is extremely fast, reaching the millisecond level, which fully meets the needs of industrial real-time detection. EfficientAD also has high performance in accuracy, and by combining the knowledge of local and global teacher models, the model can simultaneously detect subtle texture anomalies (such as scratches, stains and loose pins) and larger structural anomalies (such as missing parts and misalignment). The training time is also very fast, because there is

doi:10.6342/NTU202503439

no need to fine-tune the backbone network. The pre-trained backbone network is frozen during the entire training process, which greatly reduces the computing resources and time required for training.

However, it is very dependent on the pre-trained model, and the performance of the model is highly dependent on the quality of ImageNet pre-trained features. If the image field to be detected is very different from the natural image of ImageNet (for example, X-rays, infrared images, microscopic images), the pre-trained features may not be optimal, which may affect the final effect. In addition, the training process is relatively complicated. The training process of EfficientAD involves three independent models (two teachers, one student) and a fixed backbone network, and the construction and management of the entire process are more cumbersome. It may also be insensitive to logical anomalies. Similar to most methods based on feature reconstruction or feature distribution modeling, it is mainly good at detecting appearance anomalies (structural, texture, color anomalies). It may not be able to effectively identify logical anomalies (for example, a screw that looks completely normal but is misplaced). Finally, it requires a large number of defective images to work properly, which requires a long collection process and has no advantage for frequently changing products.

doi:10.6342/NTU202503439

The core idea of the VAE-based Deep SVDD model is to use the structure of VAE to solve a key problem of standard Deep SVDD, namely "Hypersphere Collapse" or "Trivial Solution" problem. Standard Deep SVDD has a potential risk: when training the network to map the features of all normal samples to the center *c* of the hypersphere, if there is no constraint, the network may learn a "shortcut" - it maps any input (whether normal or abnormal) to the same point *c*. In this way, the loss of SVDD (the distance to the center) becomes extremely small, but the model completely loses its ability to distinguish and becomes useless. This is the so-called "hypersphere collapse".

VAE-based Deep SVDD effectively prevents this problem by combining the Deep SVDD objective with the VAE structure and objective function. It has a unique loss function, Hybrid Objective Function, which optimizes two objectives at the same time:

Deep SVDD loss (in latent space): This loss is similar to the standard Deep SVDD, but it works in the latent space. It requires that the latent representation $z = E(x)$ obtained by all normal samples *x* after passing through the encoder *E* has the smallest Euclidean distance to the predefined center point *c*.

$$L_{SVDD} = E[\|E(x) - c\|^2]$$

VAE reconstruction loss: that is, the decoder D is required to reconstruct the original input image x as perfectly as possible based on the potential representation *z*. The loss

doi:10.6342/NTU202503439

function is usually the Mean Squared Error (MSE) or cross entropy between the input x and the reconstructed output $x' = D(E(x))$.

$$L_{recon} = E[\|x - D(E(x))\|^2]$$

These two objective functions form a constraint and balance. The SVDD loss attempts to "press" the potential representation z of all normal data to the center point c. The reconstruction loss applies a "pull" in the opposite direction: if the encoder really projects everything to the same point c, then z will not contain any valid information about the original image x, and the decoder will never be able to reconstruct the original image, resulting in a huge reconstruction loss.

Therefore, to minimize these two losses at the same time, the encoder is forced to learn an "optimal" latent space: this space must be compact enough (satisfying the SVDD goal) and retain enough valid information to complete the reconstruction task (satisfying the VAE goal). This fundamentally avoids the "hypersphere collapse".

However, since the model introduces new hyperparameters, the model needs to make a trade-off between SVDD loss and reconstruction loss. This trade-off is usually controlled by a hyperparameter λ ($L_{total} = L_{SVDD} + \lambda * L_{recon}$). How to set the value of this λ is crucial to the performance of the model and needs to be adjusted through experiments, which increases the complexity of model debugging. Moreover, the

doi:10.6342/NTU202503439

reconstructed image will be blurry. This is because to make the latent space have good probability distribution characteristics, the reconstructed image generated by VAE is usually blurrier and smoother than the standard Auto-Encoder (AE). The most fatal disadvantage is its ability to reconstruct tiny anomalies. The generalization ability of VAE-based Deep SVDD is too strong, and even some tiny anomalies (anomalies) can be "successfully" reconstructed, which will lead to a lower reconstruction error of these anomalies, making them difficult to detect.

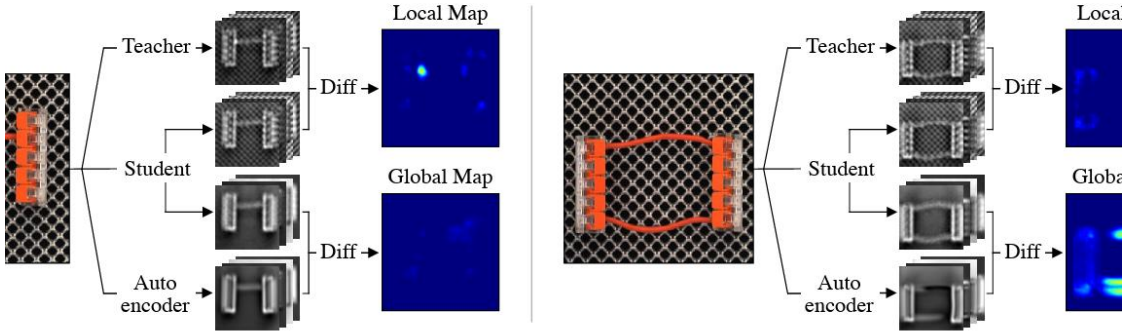


Figure 2-2-4-1: EfficientAD architecture [10].

doi:10.6342/NTU202503439

Chapter 3 Background of Methodology

3.1 Overview

Reconstruction models and conditional denoising diffusion models can be used for anomaly detection, but the reconstruction speed is too slow. We need a faster image generation method. Variational Auto-Encoders (VAEs) is a fast reconstruction method and one of the basic architectures of the ZhangInspect algorithm. So, in this section we will explore the principles of VAE and its derivation process, as well as the background reasons why VAE can perform defect detection, and its advantages in defect detection.

3.2 Deep Auto-Encoder [2]

Deep Auto-Encoder uses the powerful fitting ability of neural networks to significantly reduce the dimension of the image and restore it to a similar image to the greatest extent, which lays the foundation for reconstructing the image. For example, Figure 3-2-1 is a four-layer neural network that expands an image to 784 dimensions. Expanding the image will lose some nonlinear features. Generally, the dimension needs to be expanded to 1000 dimensions and then compressed to retain more features to prevent the gradient from disappearing. After compression to 30 dimensions, the output Code is usually called the latent space, but because this structure outputs a limited number

doi:10.6342/NTU202503439

of Codes, its output is quite discrete, which means that if the input obtained by the decoder is not among all the trained Codes, it will output unrecognizable garbled images (Figure 3-2-2). At the same time, we also hope that for similar images (for example, Piece 521 represents a class of similar images, each large piece will have a Piece 521 small piece, these Piece 521 small pieces are basically the same, with only slight differences or anomalies), the output Code can be gathered in a range, that is, similar images should not have irregular points in the latent space due to the nonlinear transformation process of the neural network. For example, we can add a little noise to the encoder to cover the distorted area and make it present a Gaussian distribution (Figure 3-2-3). In this way, for similar images such as Piece 521, meaningful images can be decoded in the entire latent space, and the closer to the probability peak, that is, the center of the Gaussian distribution, the closer the image is to a normal image, and the farther away from the center of the Gaussian distribution, the more anomalies the image will have (because anomalies are similar to noise, there is more noise in areas far from the center of the Gaussian distribution.).

doi:10.6342/NTU202503439

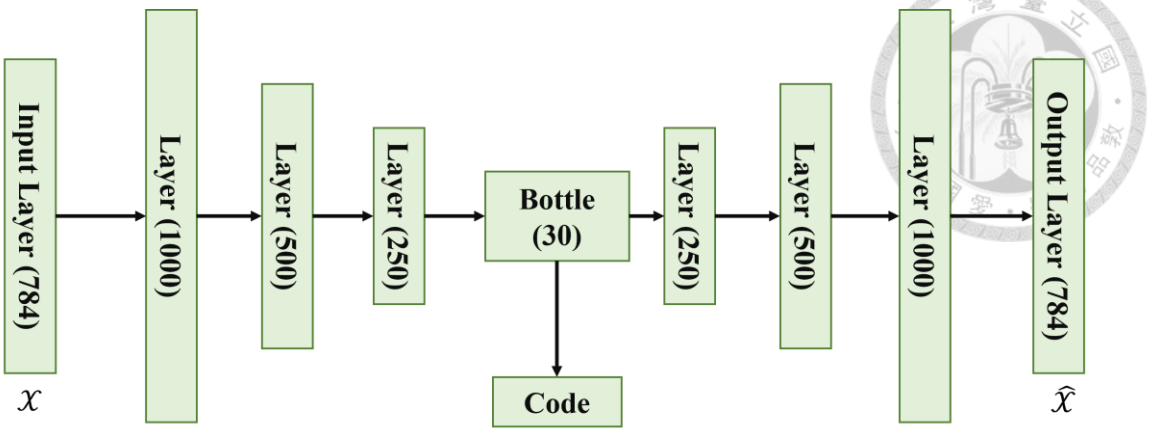


Figure 3-2-1: Deep Auto-Encoder Architecture (if the input image size is 28*28 pixels). It is able to compress a 784-dimensional vector to 30 dimensions and decode it back to a similar image. (The number in brackets is the number of neurons in the neural network, that is, the dimension of the image after expansion.)

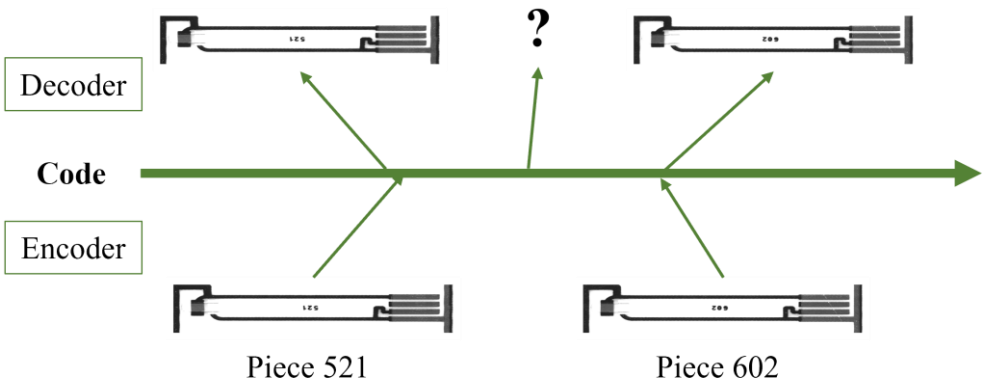


Figure 3-2-2: The discreteness of Deep Auto-Encoder will cause unknown codes to generate garbled codes. During the training process, both Piece 521 and Piece 602 have specific code values in the latent space, but the code value of the space between the two is likely to be a meaningless garbled image after decoding.

doi:10.6342/NTU202503439

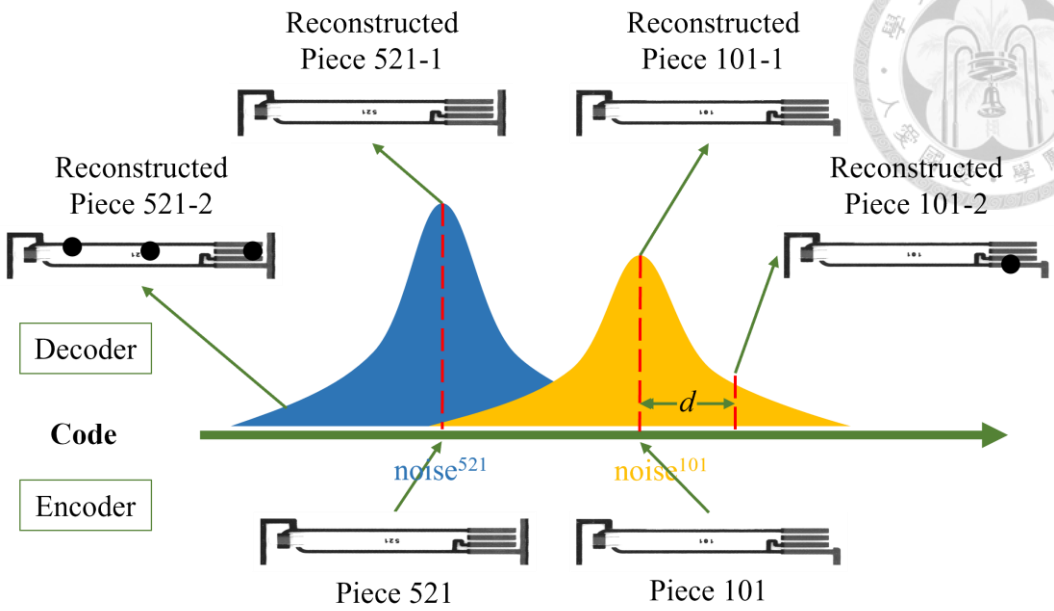


Figure 3-2-3: Deep Auto-Encoder becomes a continuous normal distribution space after adding noise. Among them, noise521 and noise101 represent the probability Gaussian distributions of different similar test pieces after adding different noises. The left side is the Gaussian distribution of Piece 521, and the right side is the Gaussian distribution of Piece 101. Although Piece 521 and Piece 101 are both small pieces in the UA large piece, they are different in shape and center number. Because their own probabilities of occurrence are different, their Gaussian distribution maximum values are also different. This allows us to decode the image at any point in the latent space. For example, Piece 101-1 and Piece 101-2 are images reconstructed from two code values in the latent space. The closer to the probability center, the fewer anomalies, that is, the smaller d, the fewer anomalies.

doi:10.6342/NTU202503439

3.3 Distribution Gaussian Mixture Model

For generative models, the mainstream theoretical models can be divided into Hidden Markov Model HMM, Naive Bayes Model NB, and Gaussian Mixture Model GMM, and the theoretical basis of VAE is Gaussian Mixture Model. By adding noise to the probability $P(m)$ of an image (e.g., Piece 521 in Figure 3-2-3), and converting it into its Gaussian distribution $P(1)$ (e.g., the blue Gaussian distribution noise521 on the left side of Figure 3-2-3), we obtain a continuous probability distribution space for the image. Next, we need to superimpose the continuous probability distribution spaces of different images to obtain a comprehensive probability space $P(x)$, as shown in Figure 3-3-1. This superposition allows us to distinguish subtle differences between small samples of images with the same Piece ID (e.g., UA), enhancing reconstruction capabilities. For example, different similar data sets represented by different numbers in the center of a small sample (e.g., Piece 521 and Piece 101 in Figure 3-2-3) can have their own probability peaks.

In simple terms, the Gaussian Mixture Model means that any data distribution $P(x)$ can be regarded as the superposition of several Gaussian distributions.

$$P(x) = \sum_{m} P(m)P(x|m)$$

However, the Gaussian mixture model is discrete, so we need to transform its input

doi:10.6342/NTU202503439

into a vector from normal distribution $z = \mathcal{N}(0,1)$, which means that the input will be continuous, and then do the integration. Then each z represents an attribute of the image, and as long as the number of z is controlled, the accuracy and structure of the image can be controlled (Figure 3-3-1). For each sample z, two functions $\mu(z)$ and $\sigma(z)$ can be decoded:

$$P(x|z) = \mathcal{N}(\mu(z), \sigma(z))$$

Determine the mean and variance of the Gaussian distribution corresponding to z.

Correspondingly, for input x, we can encode two functions $\mu'(z), \sigma'(z)$, where $q(z|x)$ can represent any distribution type, that is:

$$q(z|x) = \mathcal{N}(\mu'(z), \sigma'(z))$$

Then the accumulation of all Gaussian distributions in the integral domain becomes the original distribution $P(x)$:

$$P(x) = \int_z P(z)P(x|z)dz$$

We hope that $P(x)$ is as large as possible, which is equivalent to:

$$Maximum\ L = \sum_x \log P(x)$$

$$\log P(x) = \int_z q(z|x) \log P(x) dz$$

doi:10.6342/NTU202503439

$$\log P(x) = \int_z q(z|x) \log(\frac{P(z,x)}{q(z|x)} \frac{q(z|x)}{P(z|x)}) dz$$

$$\log P(x) = \int_z q(z|x) \log\left(\frac{P(z,x)}{q(z|x)}\right) dz + \int_z q(z|x) \log(\frac{q(z|x)}{P(z|x)}) dz$$

The right-hand side term KL diversion is:

$$\int_z q(z|x) \log(\frac{q(z|x)}{P(z|x)}) dz = KL(q(z|x) \| P(z|x)) \geq 0$$

The left term is defined as the lower bound L_b of $\log P(x)$:

$$\log P(x) \geq L_b = \int_z q(z|x) \log\left(\frac{P(x|z)P(z)}{q(z|x)}\right) dz$$

The overall formula transforms to:

$$\log P(x) = L_b + KL(q(z|x) \| P(z|x))$$

We need to find the maximum value of similarity, that is, to continuously increase the lower bound L_b. We can try to fix $P(x|z)$, because $\log P(x)$ is only related to $P(x|z)$, so we can fix $P(x|z)$ to find the maximum value of L_b. At this time, we adjust $q(z|x)$ to make L_b rise, and Kullback-Leibler Divergence (KL divergence) will decrease. If KL divergence drops to 0, $\log P(x)$ and L_b are completely consistent, but in actual situations, we do not need $\log P(x)$ to be completely consistent with L_b, which is even harmful. During anomaly detection, we need to reconstruct the image as closely as possible. However, if the reconstructed image is exactly the same as the original image,

doi:10.6342/NTU202503439

the anomaly will also be completely reconstructed, completely losing the ability to detect it (many models with strong reconstruction capabilities have this problem).

Thus, VAE means, in a macro sense, that adjusting $P(x|z)$ means adjusting the Decoder, and adjusting $q(z|x)$ means adjusting the Encoder. In each cycle, we fix the Decoder and let the Encoder get close to the Decoder to become its lower bound.

For the lower bound L_b, we will break it down:

$$L_b = \int_z q(z|x) \log\left(\frac{P(z)}{q(z|x)}\right) dz + \int_z q(z|x) \log P(x|z) dz$$

$$L_b = -KL(q(z|x)\|P(z)) + \int_z q(z|x) \log P(x|z) dz$$

We find that there is a negative KL divergence $-KL(q(z|x)\|P(z))$, which we can expand:

$$-KL(q(z|x)\|P(z)) = \int_z q(z|x)(\log(P(z)) - \log q(z|x)) dz$$

$$-KL(q(z|x)\|P(z)) = \frac{1}{2}\sum_{j=1}^{D}\left(1 + \log(\sigma_j^2) - \mu_j^2 - \sigma_j^2\right)$$

doi:10.6342/NTU202503439

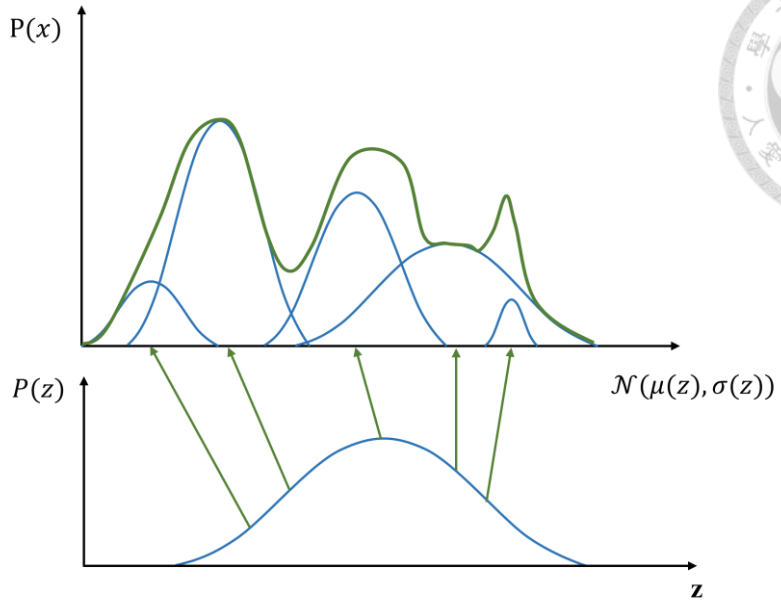


Figure 3-3-1: Continuous encoding method of VAE.

3.4 Variational Auto-Encoders (VAEs) [6]

The above formula is the main technology we use, Variational Auto-Encoder [6]. The core of VAE is to map the input data to the probability distribution (Gaussian distribution) of the latent space and generate data by sampling from this distribution. The training goal of VAE is to maximize the Evidence Lower Bound (ELBO), which is expressed as:

$$ELBO = \mathbb{E}_{q(z|x)}[\log p(x|z)] - KL(q(z|x)\|p(z))$$

which balances two objectives:

The first term is the Reconstruction Loss: Used to measure the similarity between the data generated by the decoder and the input, often using Mean Squared Error (MSE).

doi:10.6342/NTU202503439

The second term is the KL Divergence Loss (Kullback-Leibler Divergence Loss): Used to measures the difference between the latent distribution $q(z|x)$ learned by the encoder and the prior distribution $p(z)$ (usually $\mathcal{N}(0,1)$), regularizes the latent space to approximate a standard normal distribution, enabling generative capabilities.

3.4.1 KL Divergence Loss

KL Divergence (Kullback-Leibler Divergence) is a measure of the difference between two probability distributions. In VAEs, KL divergence loss is used as a regularization term to ensure that the potential distribution $q(z|x) = \mathcal{N}(\mu, \sigma^2)$ generated by the encoder is close to the prior distribution $p(z) = \mathcal{N}(0,1)$. Its main functions include:

Regularize the latent space: Prevents the latent variables from collapsing into a single representation, ensuring diversity in generated samples.

Support generative tasks: by making the latent distribution close to the standard normal distribution, the model can generate new data from $\mathcal{N}(0,1)$ samples.

Improve model stability: avoid overfitting and ensure that the latent representation is robust to changes in the input data.

Figure 3-4-1-1 shows that the prior distribution of the latent space is Gaussian.

doi:10.6342/NTU202503439

Unlike standard autoencoders, where the decoder's input can only come from the corresponding encoder's output without an inter-sample distribution, AutoencoderKL, based on a continuous latent space representation, can generate a continuous distribution between different samples.

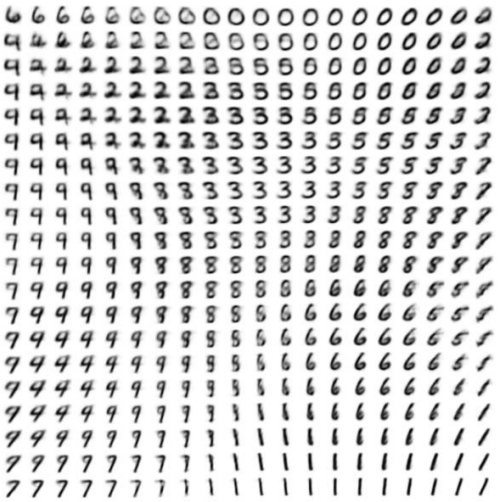


Figure 3-4-1-1: Illustration of the Continuous Distribution of the Latent Space in AutoencoderKL [6].

doi:10.6342/NTU202503439

3.4.2 Mathematical Derivation of KL Divergence Loss

For a potential dimension *j*, assume that the encoder outputs a Gaussian distribution $q(z_j|x) = \mathcal{N}(\mu_j, \sigma_j^2)$, and the prior distribution is $p(z_j) = \mathcal{N}(0,1)$. The KL divergence:

$$KL\left(q(z_j|x)\|p(z_j)\right) = \int q(z_j|x) \log \frac{q(z_j|x)}{p(z_j)} dz_j$$

For Gaussian distribution, KL divergence has an analytical solution:

$$KL\left(q(z_j|x)\|p(z_j)\right) = -\frac{1}{2}\int \left(1 + \log(\sigma_j^2) - \mu_j^2 - \sigma_j^2\right)$$

where $\sigma_j^2 = \exp(logvar_j)$. For the entire latent space (of dimension *D*), assuming that each dimension is independent, the KL divergence is:

$$KL\left(q(z_j|x)\|p(z_j)\right) = -\frac{1}{2}\sum_{j=1}^{D}\left(1 + \log(\sigma_j^2) - \mu_j^2 - \sigma_j^2\right)$$

For batch data (batch size *N*), the KL divergence loss is usually summed or averaged over all samples and dimensions.

doi:10.6342/NTU202503439

Chapter 4 Methodology

4.1 Overview

The principle of VAE anomaly detection can be simply understood as follows: for an input image, VAE will imitate the image in the training process, and the high-dimensional vector z generated after encoding will shift to the probability peak of the curve. From the VAE continuous encoding method analyzed in the previous chapter (Figure 3-3-1), its learning will generate a probability curve $P(x)$ based on Gaussian distribution (one-dimensional in the figure but high-dimensional in actual training). Due to the clustering of similar images in the probability curve in the latent space and the reduction of anomaly the closer to the probability peak, our input image will always slide towards the probability peak of its similar images in the latent space, that is, the nearest probability peak.

This is reflected in the VAE reconstruction of the image, which will repair the anomalies in the image. Because we continuously build a high-dimensional Gaussian distribution probability space of normal images during training, when we input a defective image, this defective image will enter this high-dimensional probability distribution space and shift to the highest probability point of this space (Figure 4-1-1).

doi:10.6342/NTU202503439

The probability space we trained is continuous and linear, and it is also a continuous change process in generating images. For example, our output dimension is 16 dimensions. If we choose one dimension to change the output value, the feature represented by this dimension will also change linearly.

In the macroscopic manifestation, the reconstructed image repairs the edge of the anomaly, which allows us to define the location and range of the anomaly. This is also true for real generation. As shown in Figure 4-1-2, as the image approaches the probability peak, it becomes closer and closer to the normal image.

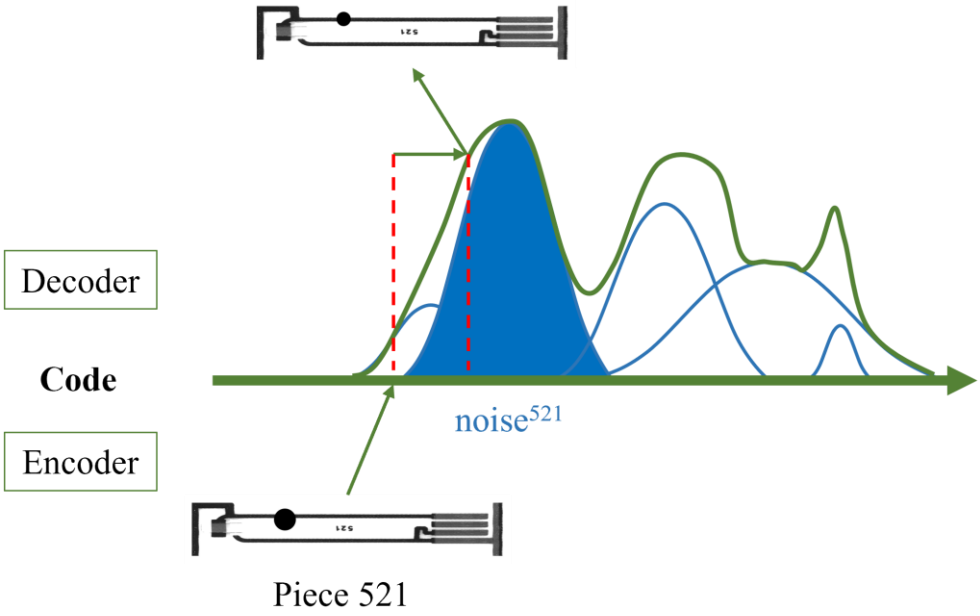


Figure 4-1-1: CVAE probability shift diagram. The input image will shift towards the probability peak and will be more like a normal image (left).

doi:10.6342/NTU202503439

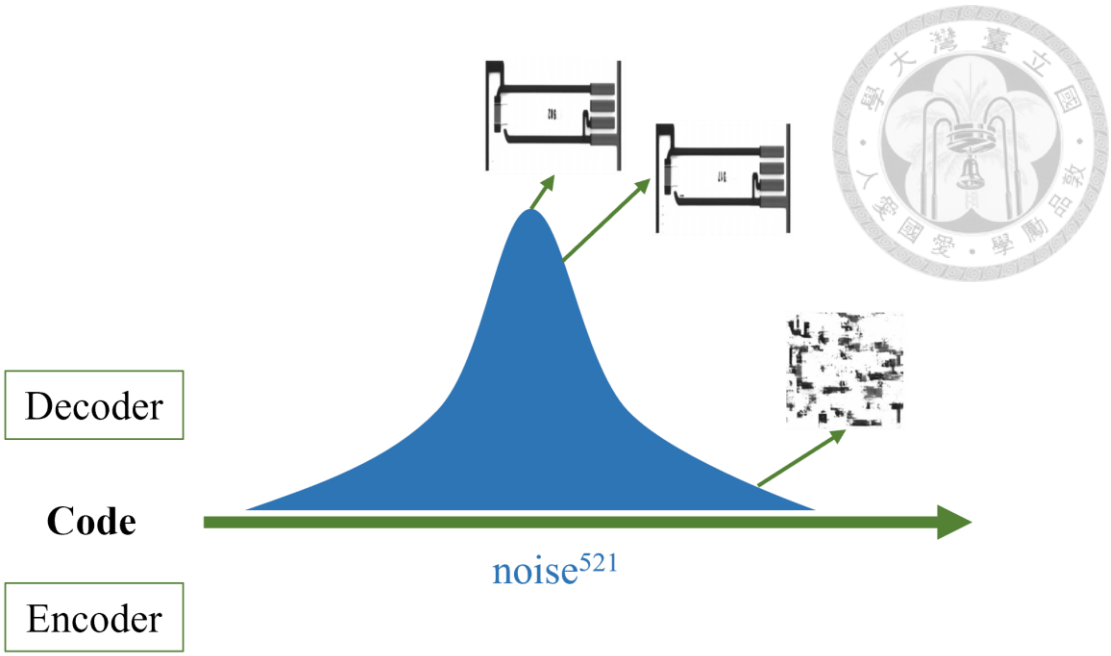
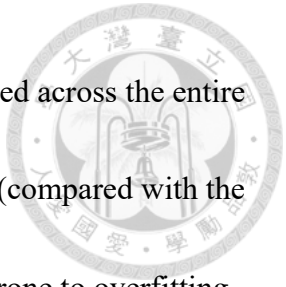


Figure 4-1-2: Latent space continuity and image restoration in CVAE. The closer to the probability peak, the less noise or anomalies the image has).

4.2 Convolutional VAE

There is a problem with the images generated by ordinary VAE, that is, the spatial structure of the generated images is poor. For example, if an error pixel appears at any point in the image, the loss score obtained is the same, which will cause great problems in anomaly detection, because the difference caused by the different positions of error pixels in anomaly detection is very large, so we need to learn the difference when the error falls at different points, that is, to strengthen the spatial structure of the model. If we introduce the concept of Convolutional Neural Networks (CNN) into VAE training and replace the original Fully-Connected Networks, then the spatial structure of the model

doi:10.6342/NTU202503439



can be improved and the convolution kernel weights of CNN are shared across the entire image, which greatly reduces the number of parameters of the model (compared with the fully connected network), making the model easier to train and less prone to overfitting.

Thus, we propose to use Convolutional VAE to train the reconstruction model for anomaly detection. Convolutional Variational Auto-Encoder (CVAE) is a generative model that combines Convolutional Neural Network (CNN) and Variational Auto-Encoder (VAE), so it combines the probability modeling ability of VAE and the spatial feature extraction ability of CNN. It consists of an encoder, a latent space, and a decoder. The encoder and decoder use convolutional layers and transpose convolutional layers, which can effectively capture local patterns and spatial hierarchical structures in images and are more efficient when processing high-dimensional image data. Convolutional layers and transpose convolutional layers are designed to process and maintain the spatial structure (height and width) of the data. Therefore, throughout the network, we maintain the tensor shape of [*C, H, W*]. Flattening is only required when it needs to be fed into a fully connected layer.

doi:10.6342/NTU202503439

4.2.1 Structure of CVAE

The architecture of CVAE usually includes the following three main parts:

Encoder: The encoder is responsible for compressing the input image into a probability distribution in the latent space, and outputs the mean and variance $\mu'(z), \sigma'(z)$ in the form of [*C, H, W*] tensors with spatial structure. The convolution layer uses multiple convolution layers (Conv2D) to extract features, with a normalization layer to standardize each channel to maintain numerical stability, accelerate convergence, and reduce gradient vanishing/explosion. Use a 2-stride convolution layer to replace the pooling layer to facilitate parameter sharing and make training more efficient.

Latent Space: The latent space is the core of CVAE. The latent vector z is sampled from the distribution defined by the mean and variance through the reparameterization trick:

$$z_i = \exp(\sigma_i) \times e_i + m_i$$

This sampling process introduces randomness, allowing the model to learn a continuous and structured latent space.

Decoder: The decoder converts the latent vector z back to the image space. The transposed convolution layer with a stride of 2 restores the output size to the input size and keeps the same connection method.

doi:10.6342/NTU202503439

4.2.2 Loss Function

Compared with traditional VAE, CVAE also needs to optimize reconstruction loss and KL loss. The loss function algorithm of traditional VAE is as follows [6]:

$$\mathcal{L}(\theta, \emptyset, x) \cong \frac{1}{2D}\sum_{j=1}^{D}\left(1 + \log\left(\sigma_j^2\right) - \mu_j^2 - \sigma_j^2\right) + \frac{1}{L}\sum_{l=1}^{L}\log p_\theta(x_l|z_l)$$

We need to find the minimum value of $\mathcal{L}$ to optimize the target. However, the problem of loss scale mismatch is prone to occur in the traditional loss calculation function. Because the reconstruction loss $\frac{1}{L}\sum_{l=1}^{L}\log p_\theta(x|z)$ is averaged over all pixels, while the KL loss $\frac{1}{2}\sum_{j=1}^{D}\left(1 + \log\left(\sigma_j^2\right) - \mu_j^2 - \sigma_j^2\right)$ is averaged over all data points, which will cause the reconstruction loss to vary with the input image. For example, if our input image size is [3,256,256], then the grave term of the reconstruction loss is 200,000 times that of the KL loss. To minimize the total loss, the optimizer will almost completely ignore the gradient of the reconstruction loss, resulting in training failure. Many successful VAE implementations (including the subsequent code of the original paper) seem to work well with mean. They usually introduce a manually weighted hyperparameter β (beta), which is the idea of β-VAE:

$$Loss = recon_{loss} + \beta * kl_{loss}$$

By carefully adjusting the value of β (for example, setting it to 0.001), we can manually bring the scale of kl_{loss} back to a level comparable to $recon_{loss}$. However,

doi:10.6342/NTU202503439

this means that every time we change the dataset, we need to find a β to optimize the training results, which is very time-consuming.

4.2.3 Sum Square Error (SSE)

Thus, we use Sum Square Error (SSE) to calculate the reconstruction loss. In the field of image reconstruction for anomaly detection, in VAE, the input items will be expanded into one dimension, and there is no problem using MSE, but if the input items of CVAE are multi-dimensional and the weights need to be adjusted to do a good job of detection, SSE essentially finds a way to "automatically" balance the scales of the two, and its effect is equivalent to using mean and finding an excellent β value:

$$\mathcal{L}(\theta, \emptyset, x) \cong \frac{1}{2}\sum_{j=1}^{D}\left(1 + \log\left(\sigma_j^2\right) - \mu_j^2 - \sigma_j^2\right) + \sum_{l=1}^{L}\log p_\theta(x|z)$$

doi:10.6342/NTU202503439

4.3 ZhangInspect

ZhangInspect is a CVAE-based anomaly detection algorithm for glucose enzyme test strips, which is used to replace the AOI anomaly detection algorithm Z-AOI for anomaly detection of industrial assembly line machines. It has the characteristics of fast training, fast reconstruction and high accuracy. Among the reconstruction-based anomaly detection algorithms, compared with common reconstruction algorithms such as the DDAD model, ZhangInspect is very effective in the field of anomaly detection of glucose enzyme test strips. In the GPU GeForce RTX 3080 (Table 4-3-1), using Z-UA dataset training and testing, its single epoch training speed is 23 times that of the DDAD model, the image reconstruction speed (image size 3*256*256) is 70 times that of the DDAD model, and the detection accuracy (Pixel PRO) is 11.2% higher than that of the DDAD model. However, it should be noted that ZhangInspect only needs 50 training epochs to achieve this effect, but DDAD fails to converge at epoch=50, and it takes epoch=1000 to achieve a detection effect close to that of ZhangInspect, and it also requires 0.38 hours of domain adaptation. Therefore, the actual training speed of ZhangInspect is 482 times that of the DDAD model, and the model training can be completed in less than 10 minutes.

doi:10.6342/NTU202503439

Table 4-3-1: Comparison of training times, reconstruction times and pixel pro for different methods. (↑ indicates that ZhangInspect is more effective.)

	Training Time(s/epoch) ↑	Reconstruction Time(ms/image) ↑	Pixel PRO ↑
Z-AOI	NAN	16.0	NAN
DDAD (epoch=1000)	137.5	132.2	85.9
ZhangInspect (epoch=50)	**5.9**	**1.9**	**95.5**

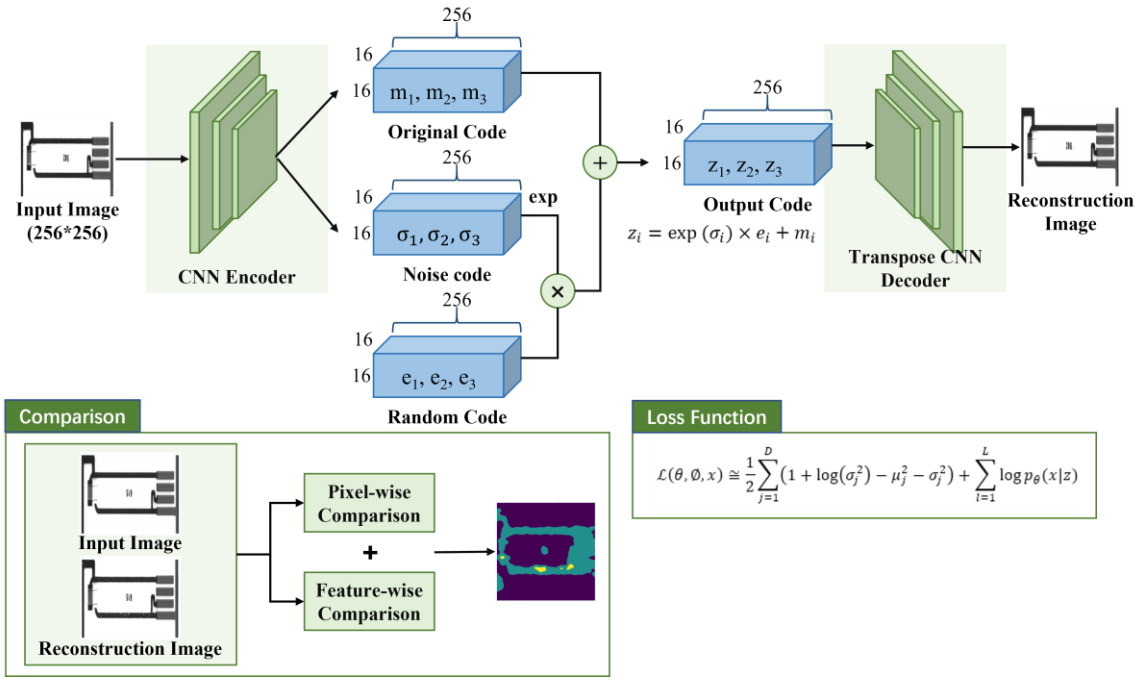


Figure 4-3-1: ZhangInspect architecture.

doi:10.6342/NTU202503439

Chapter 5 Experimental Results

5.1 Overview

This section will first discuss the datasets we use and their sources, then we will discuss the training and test results, and compare them with the DDAD model. Finally, we will discuss the evaluation methods used, Area Under the ROC Curve (AUC) and Per Region Overlap (PRO), and explain why they are used. Of course, at the end of this section, we will also attach a chart of our test results for your reference.

5.2 Datasets

The model training uses two datasets we collected ourselves: Z-UA and Z-C2B. The piece type of Z-UA dataset is the latest "Connected" type of test paper (Figure 1-2-2), and its piece ID is UA (Figure 1-2-4). It has a total of 27 large test papers, each with 150 small test papers, totaling 4050 small test papers. The scanning, splicing, cutting and marking of the test papers are completed by the Z-AOI algorithm, and the anomaly samples and their corresponding anomaly masks are output (Figure 5-2-1). It contains 68 missing ink small test papers, 3 blue pen lines small test papers, 17 both missing ink and blurred ink small test papers, 142 blurred ink small test papers, and 3820 normal small test papers.

doi:10.6342/NTU202503439

We randomly select 3520 normal small test papers for training, and the rest of the test papers are used for testing. The image size of each small test paper is 3*2020*430. In the ZhangInspect algorithm, we will convert it to a 3*256*256 image for subsequent training. The image is actually black and white, but in order to retain its color image training capability, we convert the original single-channel image into a three-channel one, and its detection speed will not be significantly affected.

The piece type of the Z-C2B dataset (Figure 5-2-2) is "Not Connected" (Figure 1-2-3), and its piece ID is C2B (Figure 1-2-4). There are a total of 93 large test strips, each with 150 small test strips, a total of 13950 small test strips, and the image size of each small test strip is 3*1850*350. The dataset was made in the same way as Z-UA, but this batch is a mature product with fewer anomalies and no missing ink or blurred ink. It includes 286 missing ink test papers, 32 blue pen lines test papers, 19 blurred ink test papers, and 13,613 normal test papers. We randomly selected 12,713 normal test papers for training, and the rest were used for testing.

The characteristics of the two datasets are that the Z-UA dataset has a smaller number of test pieces, so less training and detection time can be spent, and its anomaly has a large area and obvious features, and also a relatively complex structure, which is conducive to anomaly recognition. The Z-C2B dataset has a larger number of test pieces,

doi:10.6342/NTU202503439

and its anomaly has a small area and unclear features, but a relatively simple structure, so the reconstruction speed will be faster.

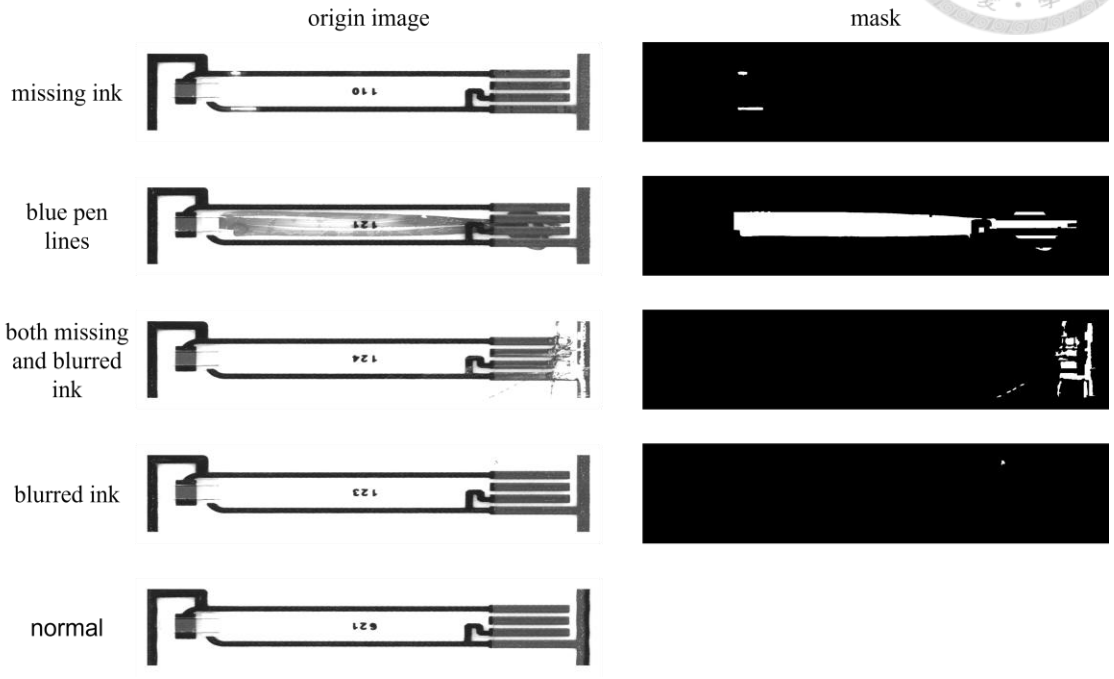


Figure 5-2-1: Z-UA dataset.

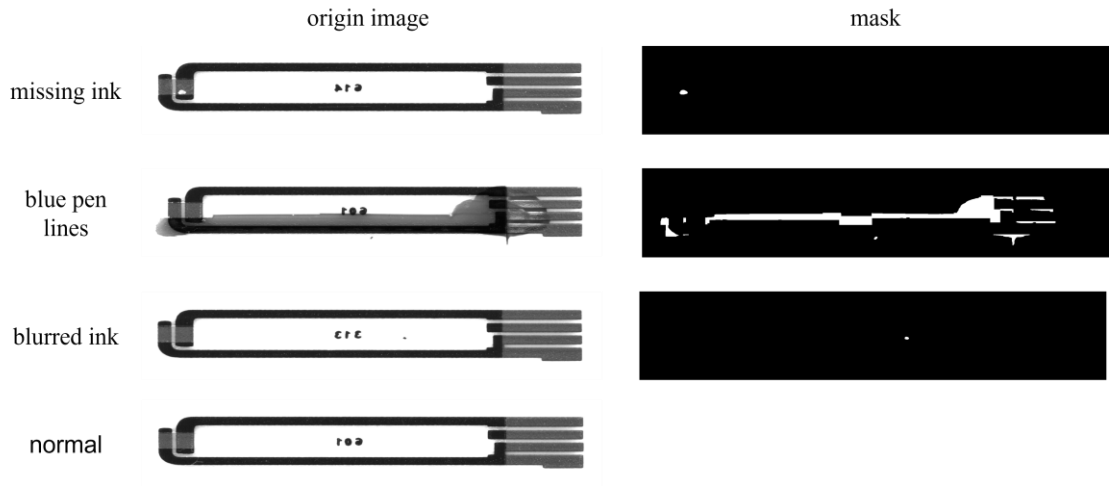


Figure 5-2-2: Z-C2B dataset.

doi:10.6342/NTU202503439

5.3 Evaluation Metric

5.3.1 Receiver Operating Characteristic Curve (ROC Curve)

The ROC curve graphically shows the performance of a binary classification model under all possible thresholds. It has two axes:

Y-axis: True Positive Rate (TPR), also known as Sensitivity or Recall. In anomaly detection, the question it answers is, among all truly defective products, what proportion of them has your model successfully identified? We hope that this indicator is as high as possible, which means that the model has a strong ability to "catch" anomalies.

X-axis: False Positive Rate (FPR), in anomaly detection, the question it answers is, among all completely normal products, what proportion of them has your model mistakenly marked as defective? We hope that this indicator is as low as possible, which means that the model has fewer "false positives" or "false alarms".

We can plot the ROC curve for ZhangInspect (Figure 5-3-1-1) by setting the threshold for the detection model using the Z-UA dataset. As shown in Figures 5-3-3-1, 5-3-3-2, 5-3-3-3, and 5-3-3-4, ZhangInspect converges faster and achieves better results on the pixel-level ROC curve, while the DDAD model performs better on the image-level ROC curve. This is due to limitations of the datasets themselves. Due to historical reasons,

doi:10.6342/NTU202503439

the datasets may have mislabeled some normal samples as anomalies and some anomalies as normal, which reduces the performance of the DDAD and ZhangInspect models on the image-level ROC curve. Therefore, using the pixel-level ROC curve as a more objective indicator of model performance demonstrates the model's ability to locate anomalies.

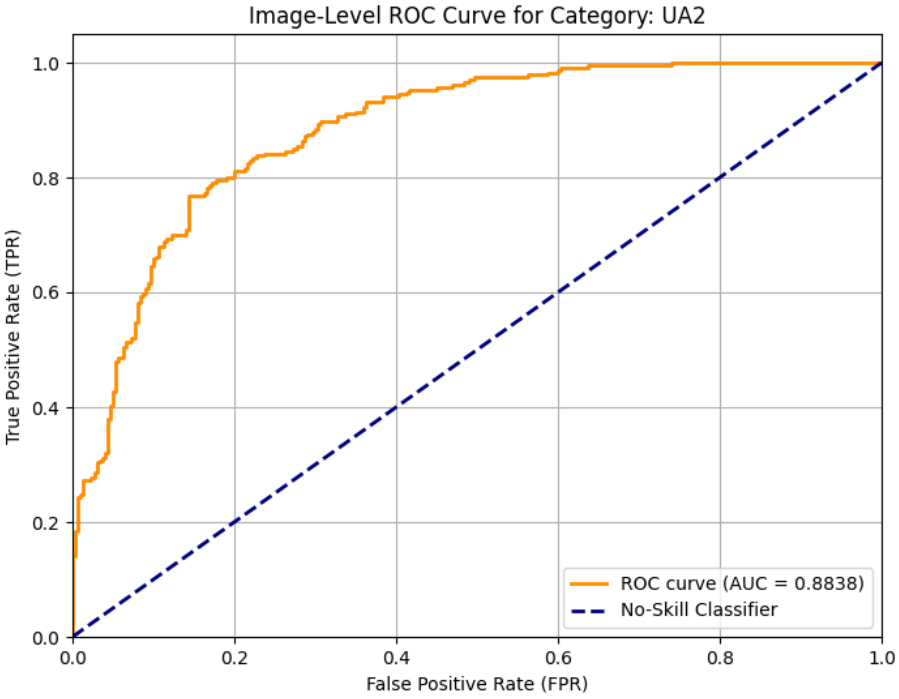


Figure 5-3-1-1: ZhangInspect Image ROC Curve example.

doi:10.6342/NTU202503439

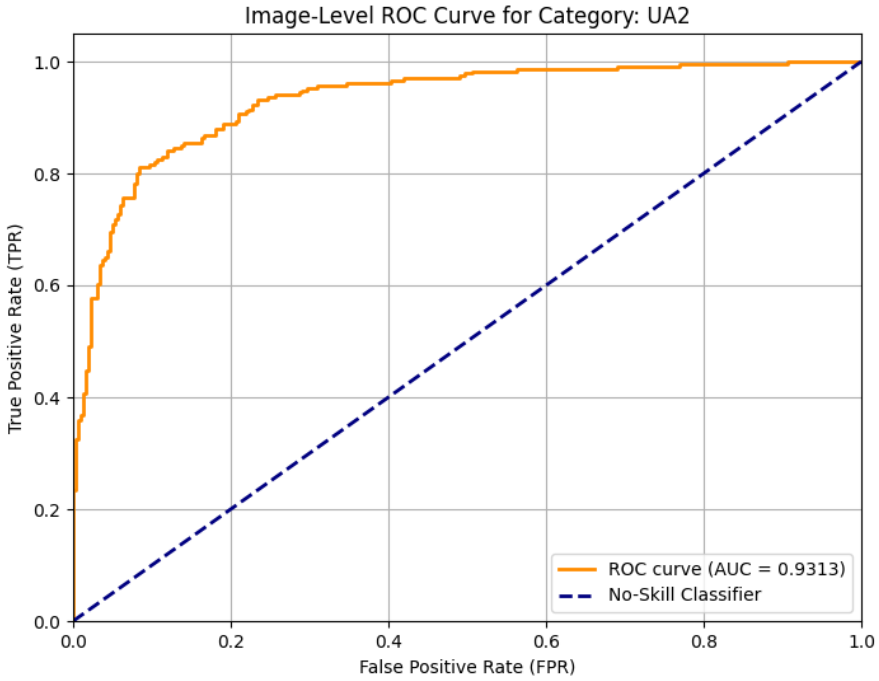


Figure 5-3-1-2: DDAD Image ROC Curve example.

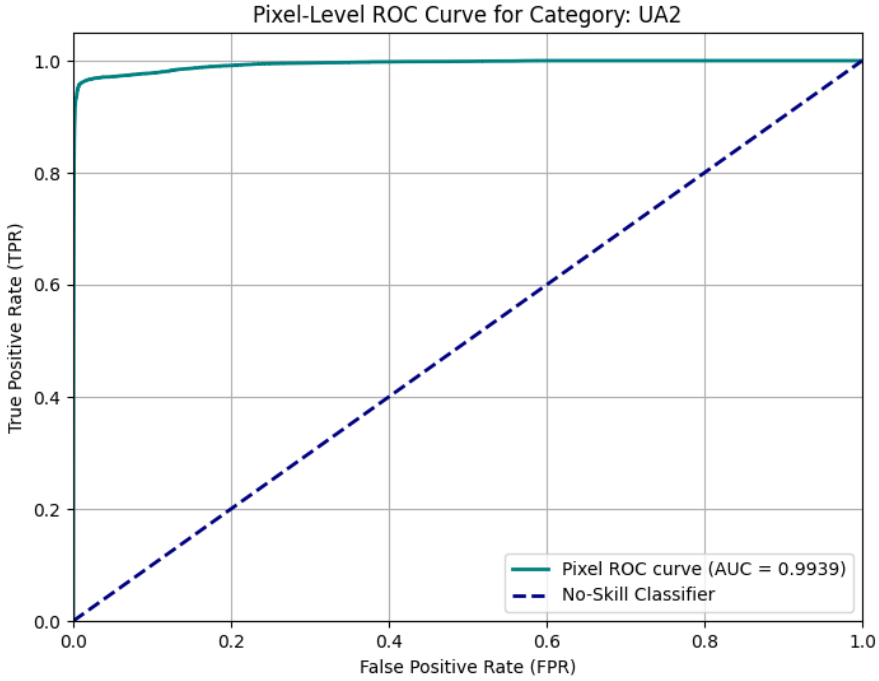


Figure 5-3-1-3: ZhangInspect Pixel ROC Curve example.

doi:10.6342/NTU202503439

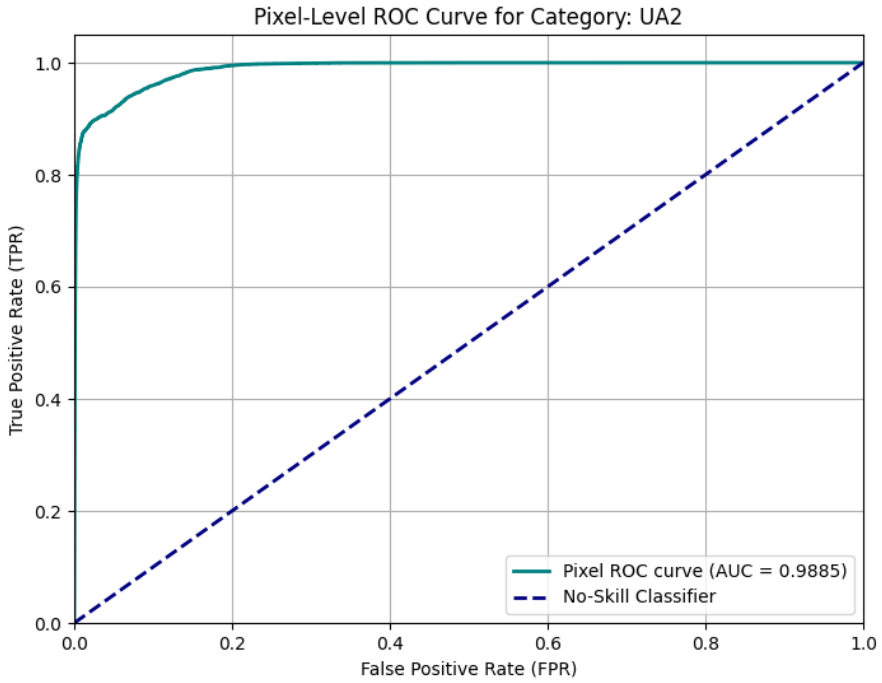


Figure 5-3-1-4: DDAD Pixel ROC Curve example.

5.3.2 Area Under the ROC Curve (AUC)

AUC is a comprehensive indicator used to evaluate the performance of binary classification models. In anomaly detection, it measures the overall ability of the model to distinguish "normal images" from "abnormal images".

It is calculated based on the ROC curve, where the horizontal axis is the false positive rate (FPR), that is, "the proportion of all normal samples that are incorrectly judged as abnormal"; the vertical axis is the true positive rate (TPR), that is, "the proportion of all abnormal samples that are correctly judged as abnormal".

doi:10.6342/NTU202503439

By moving the classification threshold from high to low on the anomaly score output by the model, a series of (FPR, TPR) points can be obtained, and connecting these points constitutes the ROC curve. AUC is the area under this curve.

AUC = 1.0: perfect classifier.

AUC = 0.5: random guessing.

AUC > 0.5: the classifier is better than random guessing.

For the results of the glucose enzyme test strip anomaly detection algorithm, using AUC to evaluate can have a single, comprehensive value to summarize the overall classification performance of the model, and it can also facilitate direct, macro comparisons between different models or different experimental settings. The most important thing is that it is threshold-insensitive (Threshold-Independence). Because in actual applications, the threshold of the anomaly score needs to be set according to specific needs (for example, it is better to kill three thousand by mistake or never let go of one). If you only compare the accuracy under a specific threshold, the results will be very biased. AUC evaluates the average performance of the model under all possible thresholds. It measures the essential quality of the model's ranking ability, that is, the model's ability to rank abnormal samples before normal samples, which makes the comparison between models fairer and objective. Finally, AUC is also a very mature and

doi:10.6342/NTU202503439

common evaluation indicator, which is widely used in the fields of machine learning and computer vision, and is easy for everyone to understand and read.

AUC is divided into image AUC and pixel AUC. The choice depends on the needs of the application: if the goal is to detect whether the entire image contains anomalies, image AUC is more appropriate; if the abnormal area needs to be accurately located, pixel AUC is more important.

5.3.3 Per-Region Overlap (PRO) [12]

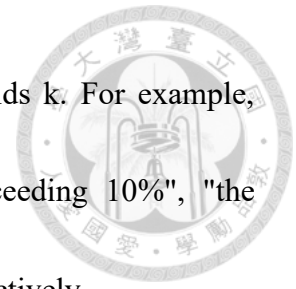
PRO is an evaluation metric designed specifically for the task of anomaly localization (or segmentation). It is more advanced than AUC because it evaluates not "whether it can determine whether there is an anomaly" but "whether the location and contour of the anomaly can be accurately found". Its calculation method is more complicated than AUC. The core idea is:

First, the pixel-level anomaly map (heatmap) output by the model is normalized and thresholder to obtain a binary predicted mask.

Then, the overlap rate between the predicted mask and the real anomaly area (ground-truth mask) is calculated, usually using the intersection over union (IoU).

The most important thing is that PRO will count the proportion of real anomaly areas

doi:10.6342/NTU202503439



that are successfully detected under different overlap rate thresholds k. For example, calculate "the proportion of overlap with the real anomaly exceeding 10%", "the proportion exceeding 20%"... "the proportion exceeding 90%" respectively.

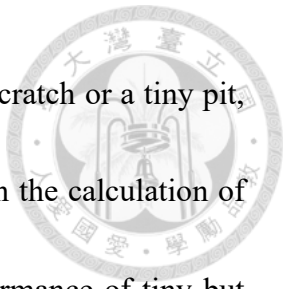
Finally, these detection proportions under different overlap rate thresholds are integrated (or averaged) to obtain the final PRO score.

A high PRO score means that the model not only finds the anomaly, but also finds the location very accurately and the contour fits well. In ZhangInspect, it has the following two advantages:

Accurate evaluation and positioning capability: This is the core advantage of PRO. In industrial applications, it is often not enough to just know that a part is "defective". We need to know exactly where the anomaly is, how big it is, and what shape it is. The PRO score directly quantifies this pixel-level positioning accuracy, which perfectly meets the actual needs of anomaly detection tasks. Reporting a high PRO score in a paper is strong evidence that your model has practical value.

Fairer to anomalies of different sizes: Traditional pixel-wise evaluation indicators (such as pixel-wise AUC) will be dominated by large areas of background. Even if a model misses very critical tiny anomalies, it may only be wrong in one ten-thousandth of the total pixels, and the indicator will still be high. PRO is evaluated "per-region", which

doi:10.6342/NTU202503439



treats each independent anomaly area equally. Whether it is a large scratch or a tiny pit, as long as it is an independent anomaly, it will be equally counted in the calculation of PRO. This makes PRO have a good measure of the detection performance of tiny but critical anomalies.

Reward accurate segmentation and penalize rough localization: If a model only outputs a blurry, large blob, although it may cover the anomaly area (IoU may not be 0), it is difficult to be considered a successful detection under a higher overlap rate threshold k. To obtain a high PRO score, the model must generate a predicted mask that is highly consistent with the true contour of the anomaly. This encourages the model to optimize towards more accurate and fine-grained segmentation.

doi:10.6342/NTU202503439

5.4 Experimental Results

The experimental results include experimental data comparison and result graph display. We conducted experiments on two self-built datasets: Z-UA and Z-C2B. Z-UA has a more complex structure but more obvious anomaly, while Z-C2B has a simpler structure but more subtle anomaly. Therefore, the result must be that Z-UA has higher detection accuracy and Z-C2B has faster reconstruction speed.

5.4.1 Comparison

ZhangInspect and DDAD are both anomaly detection algorithms based on image reconstruction, and DDAD has good performance in a wide range of anomaly detection fields, so it is particularly suitable for comparison with our model, so that the results are more objective. Our comparison will be conducted on two self-built datasets, Z-UA and Z-C2B. Table 5-4-1-1 shows the comparison of our algorithm and DDAD algorithm in detection accuracy in three evaluation methods and two data sets, using GeForce RTX 3080 graphics card. Table 5-4-1-2 shows the comparison of our algorithm and DDAD algorithm in training time and reconstruction time, using GeForce RTX 3080 graphics card. The improvement of our model in training speed and reconstruction time is very amazing, especially on the Z-C2B dataset (the flaws in this dataset are more subtle than

doi:10.6342/NTU202503439

those in the Z-UA dataset). It can be seen that our model has an exponential improvement of 523 times in training speed; 119 times in reconstruction speed, 18.5% in detection accuracy (PRO), and a small improvement in Pixel AUC. Only compression has decreased in Image AUC, but it is not worth mentioning compared with the significant improvement in other aspects.

False detection, that is, low scores on Image AUC, is unavoidable in our datasets because the labels for anomalies in our datasets are automatically generated by Z-AOI based on manual visual inspection standards.

First, Z-AOI will have a certain false detection rate, so the labels it generates and the ground truth itself will have certain errors, about 5%-15%. Therefore, manual verification of datasets will help improve the accuracy of detection, but this requires a lot of manpower.

Secondly, customers' judgment on whether it is an anomaly is inaccurate. There will be a series of thresholds, such as the size, shape and color difference of the anomaly. Anomalies close to the threshold will have a negative impact on detection, and such complex threshold types will pose challenges to detection based on image reconstruction.

Then, there will be a series of areas where anomalies exist but are not considered anomalies. This is also difficult to be learned by the reconstruction model, but will cause reconstruction errors in the reconstruction model (anomalies are learned because they are

doi:10.6342/NTU202503439

mixed in normal images and trained) (Figure 5-4-1-1).

Finally, due to the limitations of the sampling equipment design, some anomalies cannot be photographed, and some non-anomalies will be photographed (Figure 5-4-1-2).

Therefore, in terms of anomaly detection accuracy, the PRO standard is more realistic. It can define whether our model can accurately find the location and contour of the anomaly, and this ability will not be affected by the inaccuracy of the datasets. The PRO score of ZhangInspect is significantly improved compared to the DDAD model. As derived in our method, due to the characteristics of CVAE, the ZhangInspect model has a very strong ability to detect anomaly edge contours.

Table 5-4-1-1: Summarizes the results obtained by applying different methods to different datasets. (↑ indicates that ZhangInspect has stronger detection capabilities).

	Method	Image AUC	Pixel AUC ↑	PRO ↑
Z-UA	DDAD	**93.1**	98.1	85.9
	ZhangInspect	88.4	**98.8**	**95.5**
Z-C2B	DDAD	**88.9**	99.4	76.6
	ZhangInspect	83.1	**99.6**	**90.8**

doi:10.6342/NTU202503439

Table 5-4-1-2: Compares the time cost of different methods to different datasets. (↓ indicates that This means ZhangInspect's time consumption is reduced, and its training and detection speeds are fast).

	Method	Epoch ↓	Training time (s/epoch) ↓	Total Training Time (h) ↓	Testing time (ms/image) ↓
Z-UA	DDAD	1000	137.5	38.57	132.2
	ZhangInspect	**50**	**5.9**	**0.08**	**1.9**
Z-C2B	DDAD	1000	503.6	141.2	**130.7**
	ZhangInspect	**50**	**19.3**	**0.27**	**1.1**

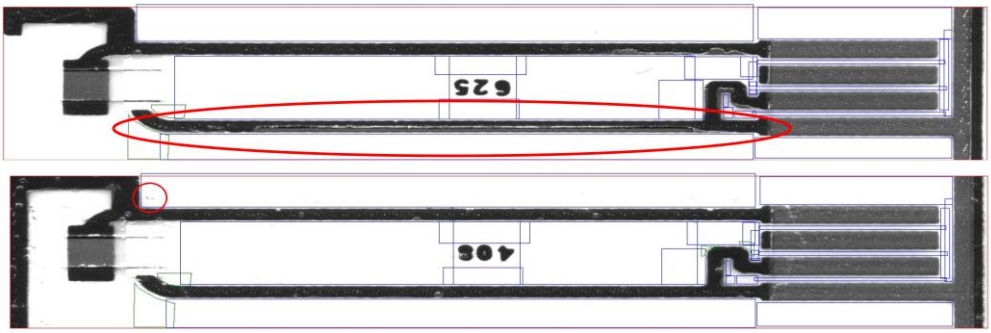


Figure 5-4-1-1: Anomalies that will be ignored in datasets.Upper picture: There are burrs on the edge of the plastic film on the test piece, which looks like missing ink, but it will be ignored. Lower picture: There are a lot of bubbles under the plastic film on the test piece, which will cause a shadow that looks like blurred ink, but it will be ignored.

doi:10.6342/NTU202503439

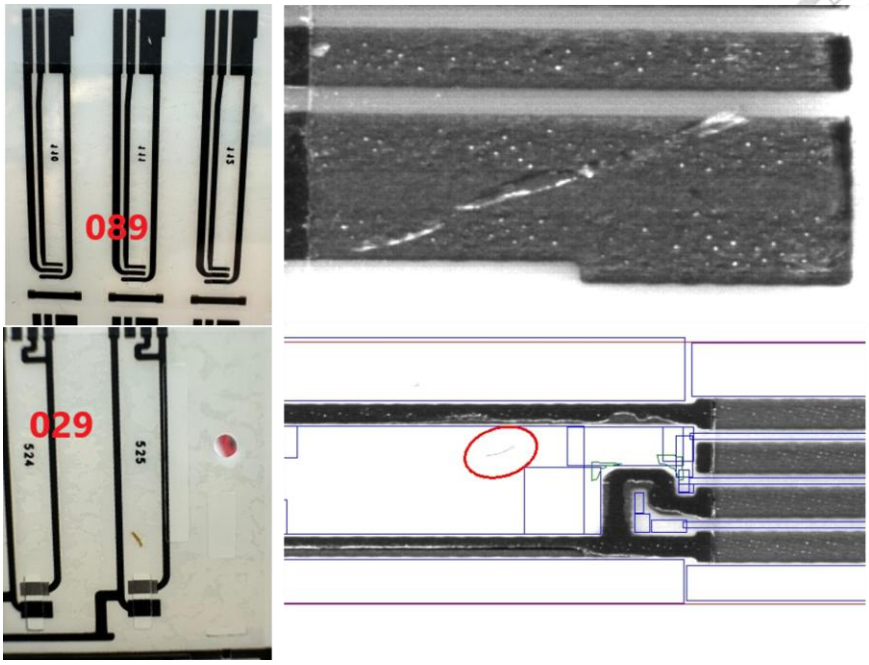


Figure 5-4-1-2: Examples of some anomalies not being detected due to machine limitations. Upper picture (The left picture is a real picture; the right picture is a picture taken by the machine camera): The missing ink anomaly cannot be captured because the test piece was completely penetrated and the black conveyor belt was photographed below. Lower picture (The left picture is a real picture, the right picture is a picture taken by the machine camera): The test piece reflected light, so the light-colored blurred ink could not be photographed.

doi:10.6342/NTU202503439

5.4.2 Z-UA Dataset Testing Results

Here we will show the representative detection results of ZhangInspect algorithm in our dataset: Z-UA. The following are some detection result images, from left to right: original image, reconstructed image, ground truth, abnormal prediction image and heat map. True Positive (TP) is the proportion of all abnormal samples that were correctly identified as abnormal. True Negative (TN) is the proportion of all normal samples that were correctly identified as normal. False Positive (FP) is the proportion of all normal samples that were incorrectly identified as abnormal. False Negative (FN) is the proportion of all abnormal samples that were incorrectly identified as normal.

- **ZhangInsepct True Negative Results:**

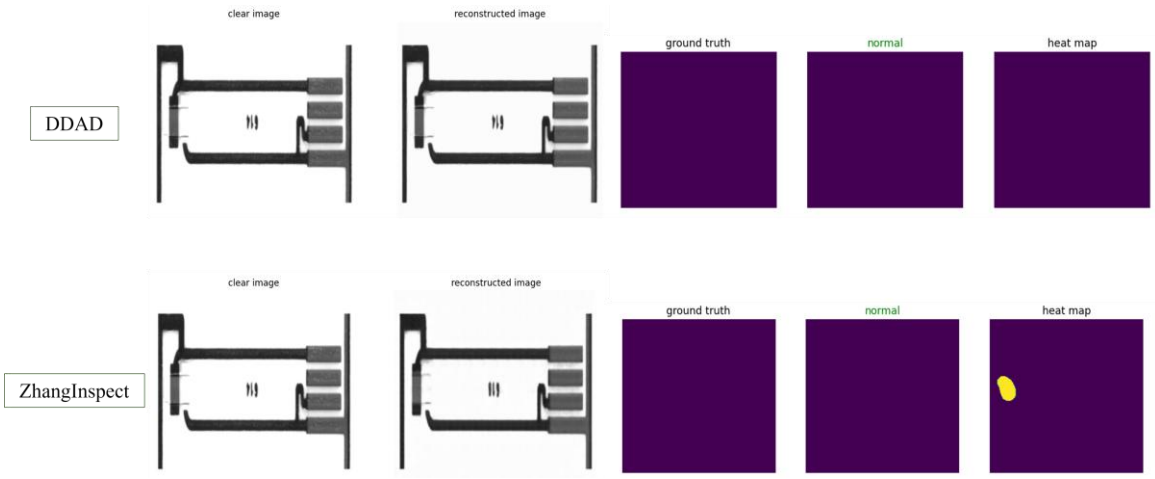


Figure 5-4-2-1: Z-UA Sample 1: Detection results: DDAD (TN), ZhangInspect (TN). All were successfully detected without any anomaly.

doi:10.6342/NTU202503439

- **ZhangInsepct True Positive Results:**

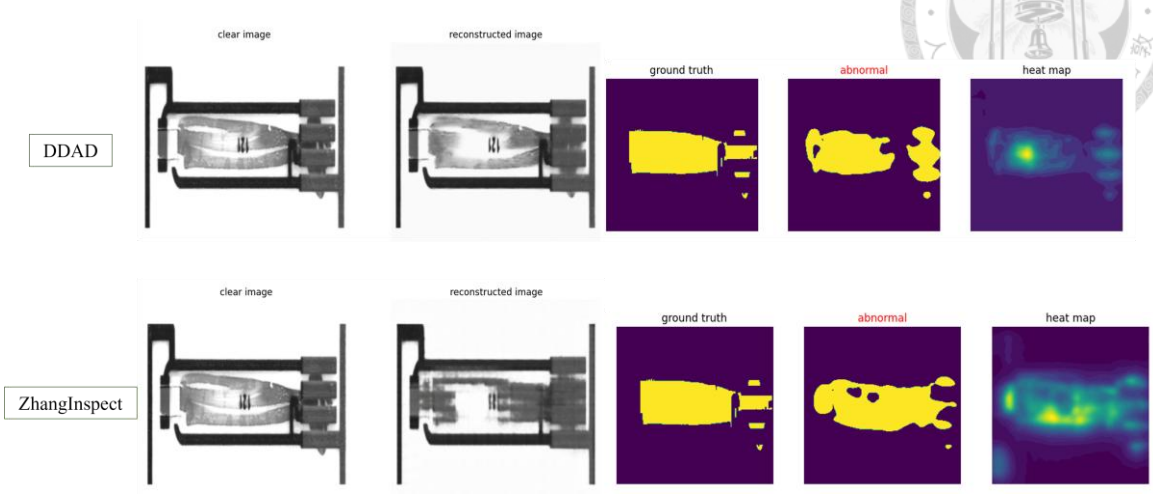


Figure 5-4-2-2: Z-UA Sample 2: Detection results: DDAD (TP), ZhangInspect (TP). Both methods detected anomalies. This is a blue pen lines anomaly. This anomaly is flawed in the mask definition because Z-AOI cannot recognize black lines in black areas in a black and white image. The clear image is the input image, and you can see that the actual anomaly is a continuous exclamation mark shape. ZhangInspect's anomaly range detection is more accurate, but due to the mask's flaws, the actual Pixel AUC score is lower.

doi:10.6342/NTU202503439

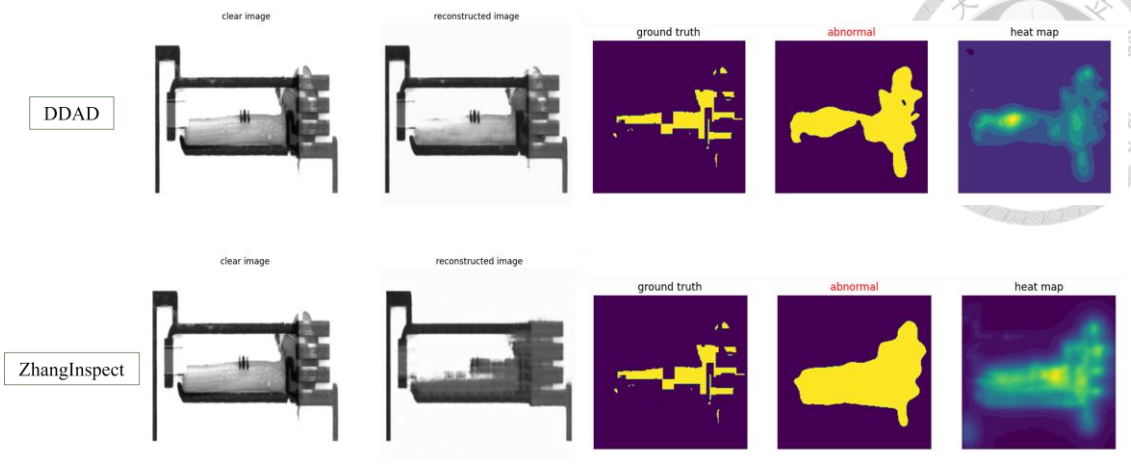


Figure 5-4-2-3: Z-UA Sample 3: Detection results: DDAD (TP), ZhangInspect (TP). Both methods detect anomalies, but ZhangInspect's defect range is more precise.

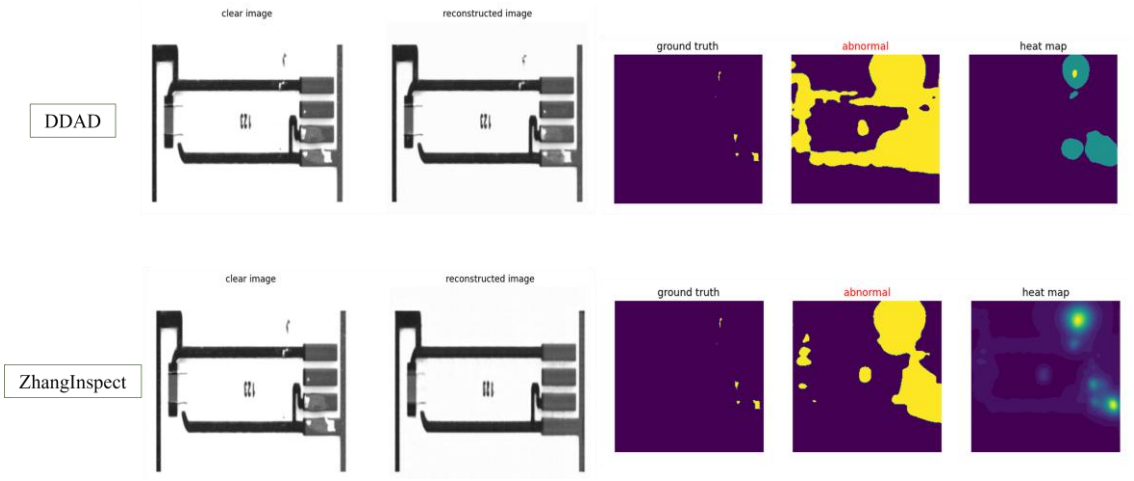


Figure 5-4-2-4: Z-UA Sample 4: Detection results: DDAD (TP), ZhangInspect (TP). Both methods detect anomalies, but ZhangInspect successfully repaired all anomalies, so the anomaly range and size are more precisely defined. However, DDAD also reconstructed the anomalies, making it impossible to accurately define the anomaly range and size.

doi:10.6342/NTU202503439

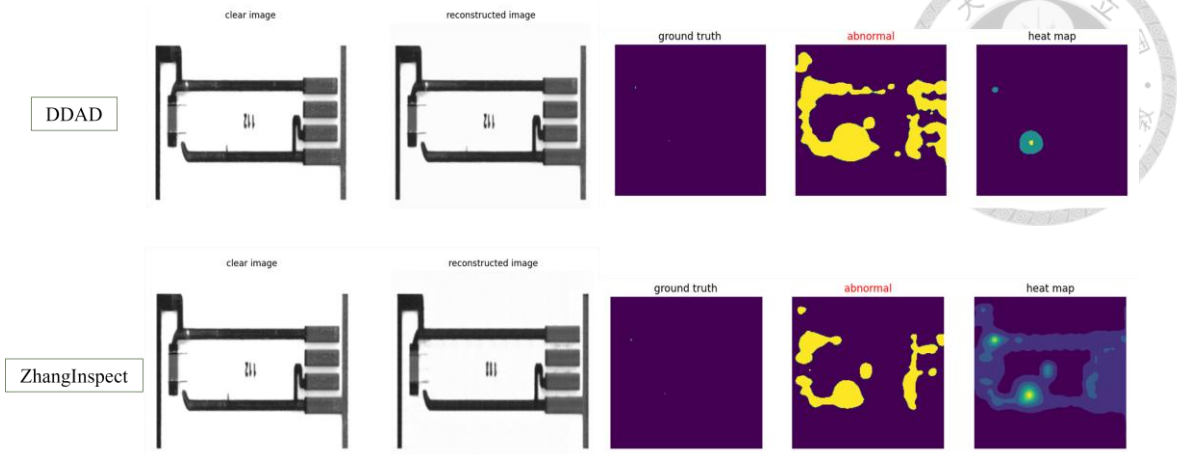


Figure 5-4-2-5: Z-UA Sample 5: Detection results: DDAD (TP), ZhangInspect (TP). Both methods detect anomalies, but it can be seen that the DDAD model reconstructs the anomaly, resulting in inaccurate definition of the anomaly's location and size. ZhangInspect successfully repaired all anomalies.

- **ZhangInsepct `False` Negative Results:**

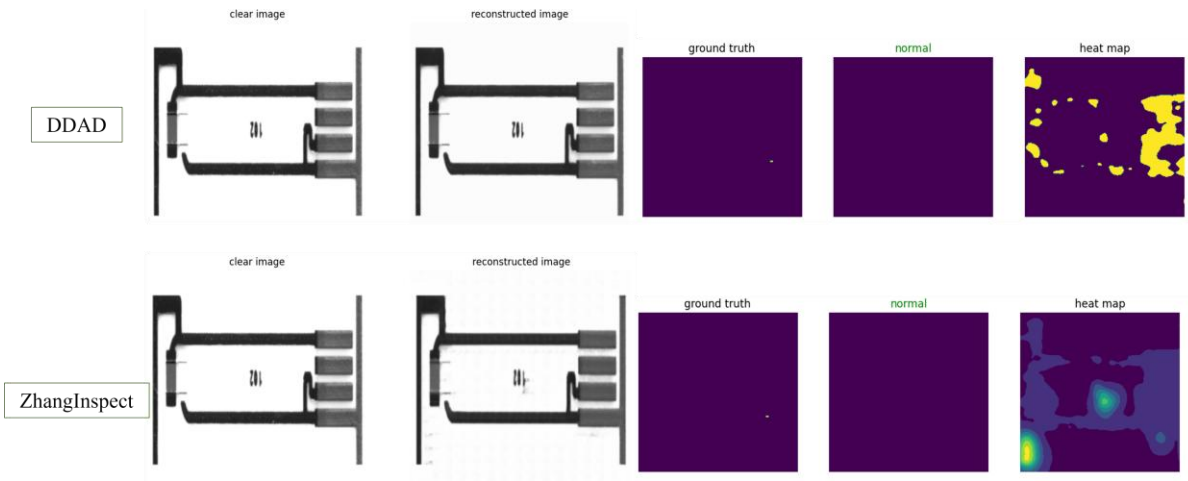


Figure 5-4-2-6: Z-UA Sample 6: Detection results: DDAD (FN), ZhangInspect (FN). Both methods are unable to identify anomalies because anomalies are very small

doi:10.6342/NTU202503439

points and can be easily misjudged.

- **ZhangInsepct `False` Positive Results:**

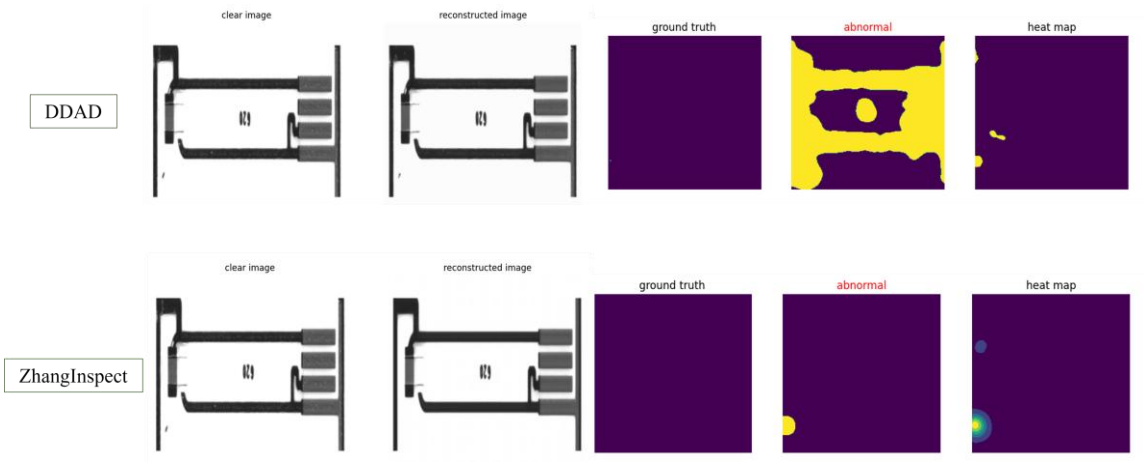


Figure 5-4-2-7: Z-UA Sample 7: Detection results: DDAD (FP), ZhangInspect (FP). Both methods incorrectly detect anomalies because the manufacturer considers this defect to be tolerable. However, this is a subjective opinion and the camera can actually capture anomalies and detect them.

doi:10.6342/NTU202503439

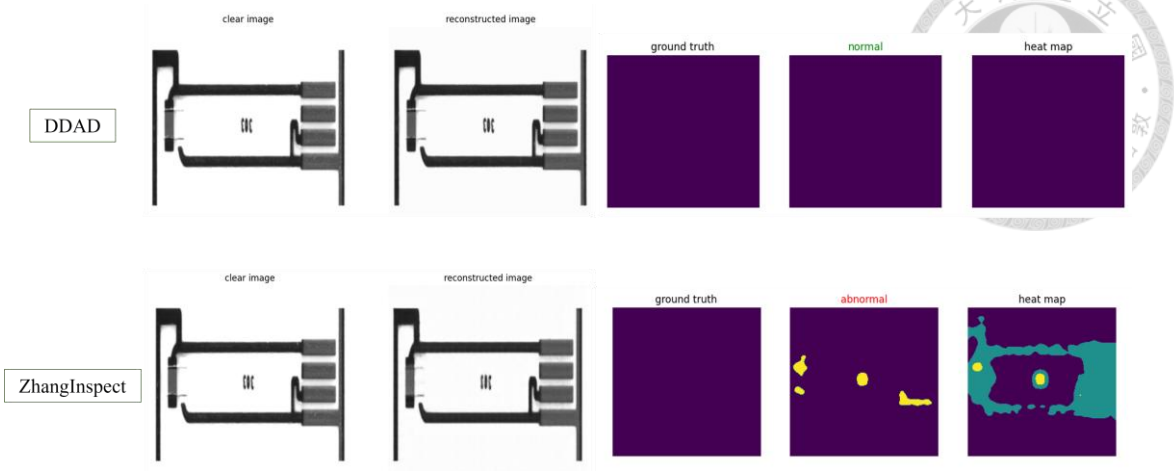


Figure 5-4-2-8: Z-UA Sample 8: Detection results: DDAD (TN), ZhangInspect (FP). This image is very typical. The anomaly area on the left side of the image is a small burr on the test piece. The anomaly type is missing ink. Because this anomaly is smaller than the threshold of Z-AOI anomaly recognition, it is marked as no anomaly, but it will be detected by ZhangInspect because ZhangInspect will try to repair all anomalies including small anomalies, but DDAD will not. The abnormal area in the middle of the image represents an error in the specimen number reconstruction process. Because the 6 and 1 are very similar, ZhangInspect identifies them as other numbers and places them within the incorrect Gaussian distribution probability peak, resulting in an incorrect reconstructed number.

doi:10.6342/NTU202503439

5.4.3 Z-C2B Dataset Testing Results

Here we show the representative detection results of ZhangInspect algorithm in our dataset: Z-C2B.

- **ZhangInsepct True Negative Results:**

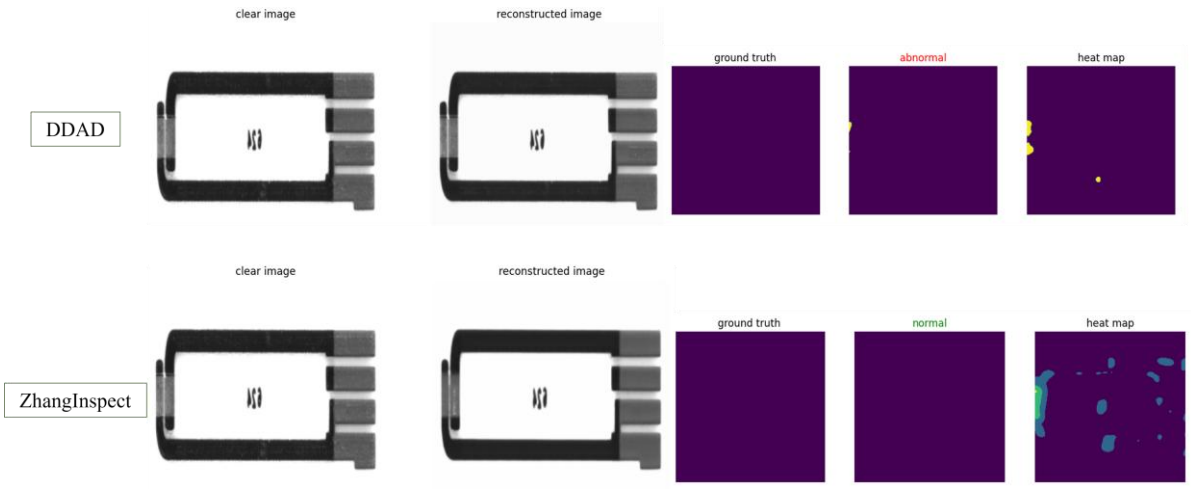
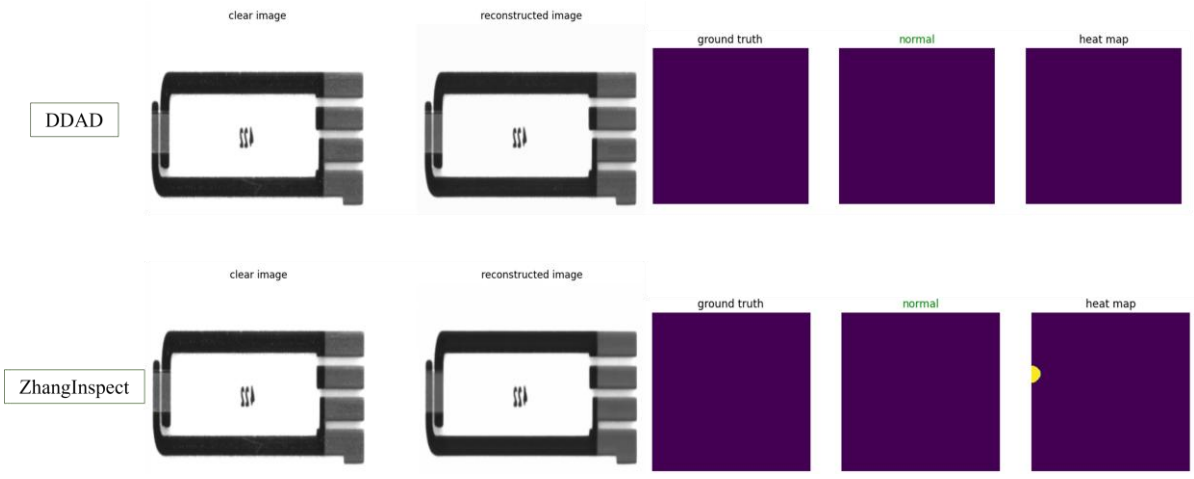


Figure: 5-4-3-1: Z-C2B Sample 1: Detection results: DDAD (FN), ZhangInspect (TN). ZhangInspect detected the defect correctly, but DDAD detected the defect incorrectly.



doi:10.6342/NTU202503439

Figure: 5-4-3-2: Z-C2B Sample 2: Detection results: DDAD (TP), ZhangInspect (TN). Both methods successfully detected no anomalies.

- **ZhangInsepct True Positive Results:**

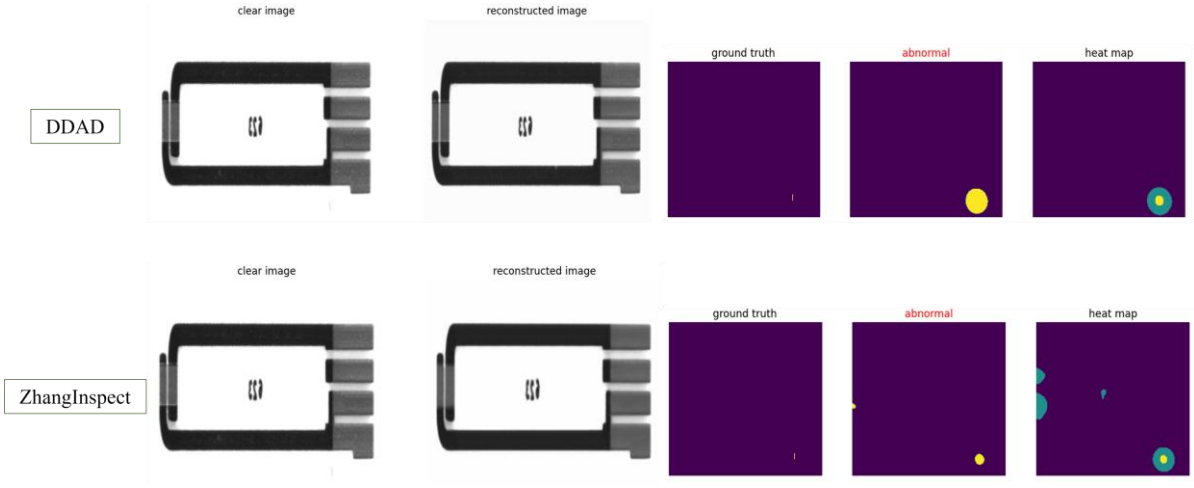


Figure: 5-4-3-3: Z-C2B Sample 3: Detection results: DDAD (TP), ZhangInspect (TP). The anomaly range defined by ZhangInspect is more precise.

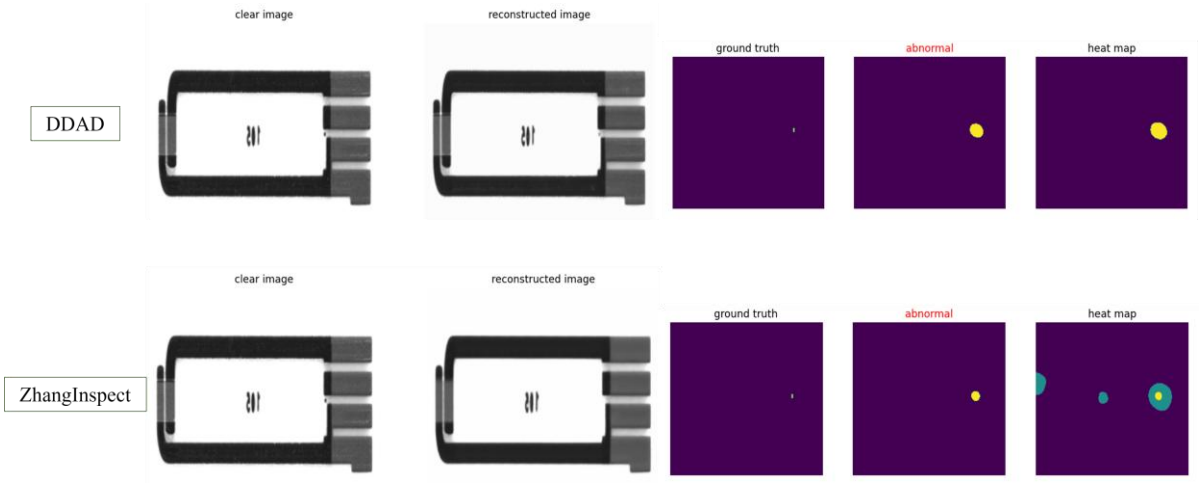


Figure: 5-4-3-4: Z-C2B Sample 4: Detection results: DDAD (TP), ZhangInspect (TP). The anomaly range defined by ZhangInspect is more precise.

doi:10.6342/NTU202503439

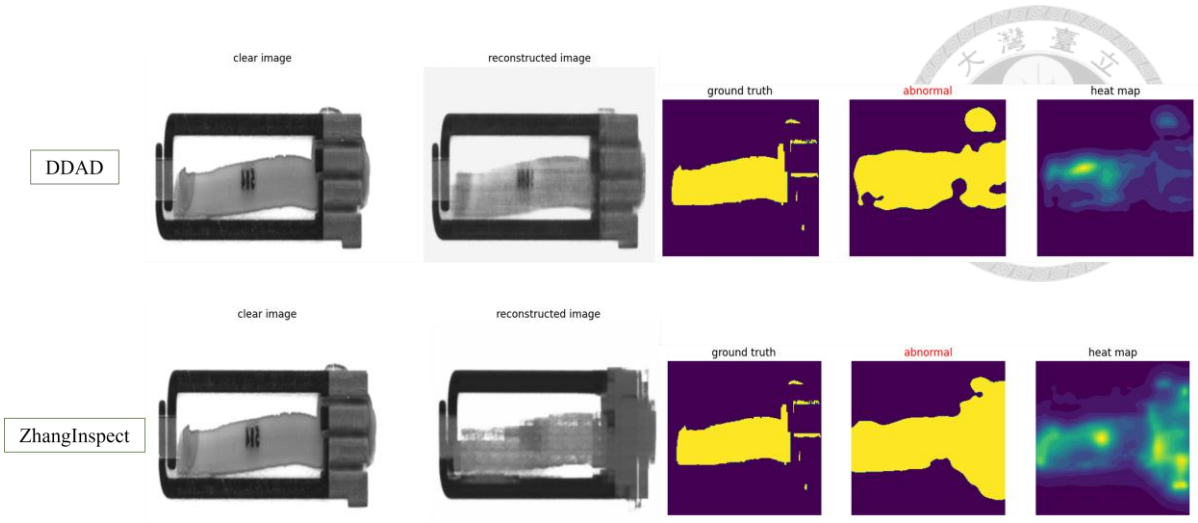


Figure: 5-4-3-5: Z-C2B Sample 5: Detection results: DDAD (TP), ZhangInspect (TP). The anomaly range defined by ZhangInspect is more precise.

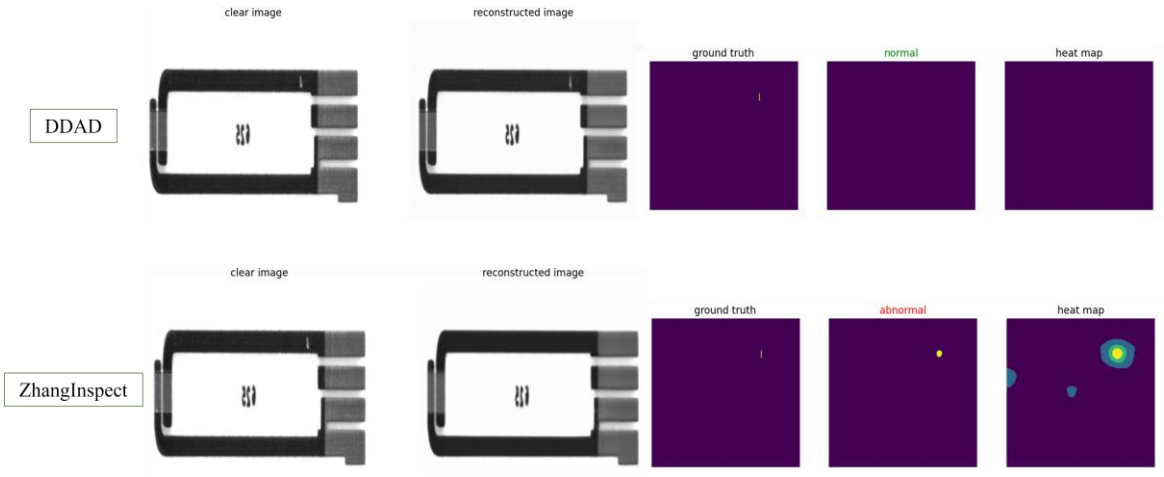


Figure: 5-4-3-6: Z-C2B Sample 6: Detection results: DDAD (FN), ZhangInspect (TP). ZhangInspect succeeded because it fixed the anomaly, but DDAD recreated the anomaly incorrectly.

doi:10.6342/NTU202503439

- **ZhangInsepct False Negative Results:**

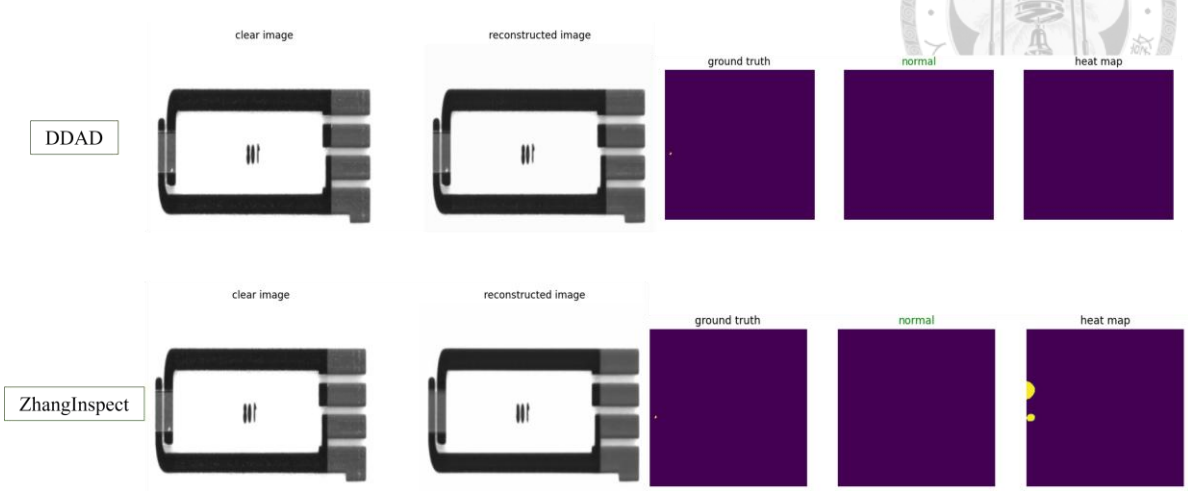


Figure: 5-4-3-7: Z-C2B Sample 7: Detection results: DDAD (FN), ZhangInspect (FN). Anomalies that are too small will be ignored. This situation is relatively rare, most of the false positives are FN.

- **ZhangInsepct False Positive Results:**

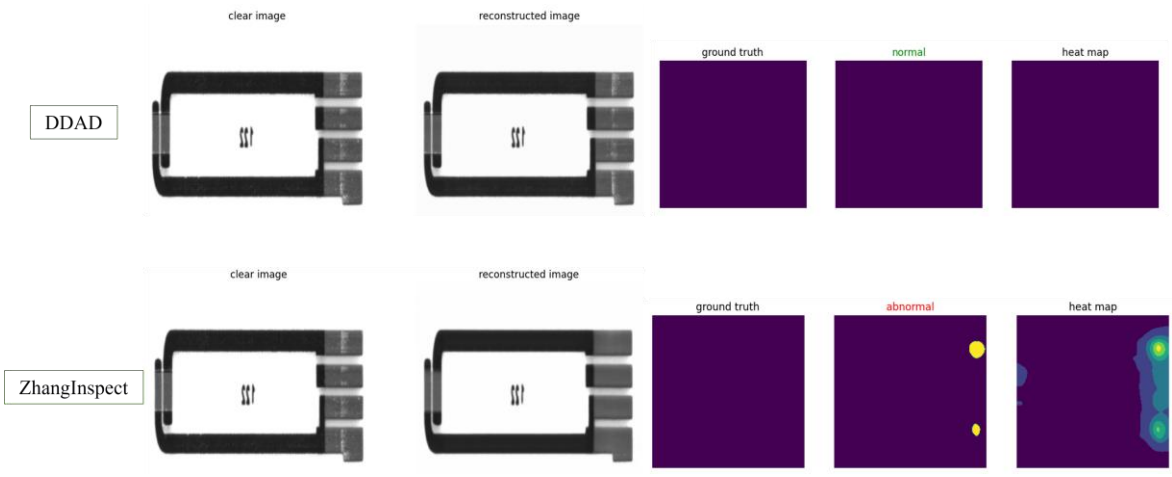


Figure: 5-4-3-8: Z-C2B Sample 8: Detection results: DDAD (TP), ZhangInspect (FP). This test piece has scratches, but they are invisible to the naked eye and require

doi:10.6342/NTU202503439

reflection to be seen, so it is judged as a normal sample. However, ZhangInspect will detect them. Our detection should be correct, but due to the limitation of the dataset, it is judged as a detection error, which reduces the Image AUC score. The DDAD model failed to detect the anomaly, but the Image AUC score is higher.

doi:10.6342/NTU202503439

Chapter 6 Conclusion and Future Works

Our ZhangInspect uses an innovative Convolutional VAE-based image reconstruction unsupervised anomaly detection model, which performs very well on the glucose enzyme datasets (Z-UA and Z-C2B), especially in terms of training speed, reconstruction speed, and anomaly edge contour detection accuracy (PRO).

During the research process, we found that the Deep SVDD algorithm is very close to our concept, and we will explore whether the Deep SVDD algorithm can be combined later. Because Deep SVDD has a potential risk: when training the network to map the features of all normal samples to the center *c* of the hypersphere, if there is no constraint, the network may learn a "shortcut" - it maps any input (whether normal or abnormal) to the same point *c*. In this way, the loss of SVDD (the distance to the center) will become extremely small, but the model will completely lose its ability to distinguish and become useless. This is the so-called "hypersphere collapse". If there is a VAE-based Deep SVDD, this problem will be effectively prevented by combining the goal of Deep SVDD with the structure and objective function of VAE. In this way, the loss function of our CVAE can be further modified to optimize two objectives at the same time:

The `first` goal is Deep SVDD loss (in latent space), which is similar to the standard

doi:10.6342/NTU202503439

Deep SVDD, but it works in latent space. It requires that the latent representation $z = E(x)$ obtained by all normal samples x after passing through the encoder E has the smallest Euclidean distance to the predefined center point c:

$$L_{SVDD} = E[\|E(x) - c\|^2]$$

The second goal is CVAE reconstruction loss. Here we can use our CVAE loss function SSE to reduce the workload of debugging weights.

These two objective functions form a constraint and balance. The SVDD loss tries to "press" the latent representation z of all normal data to the center point c. The reconstruction loss exerts a "pull" in the opposite direction: if the encoder really projects everything to the same point c, then z will not contain any valid information about the original image x, and the decoder will never be able to reconstruct the original image, resulting in a huge reconstruction loss.

Therefore, in order to minimize both losses at the same time, the encoder is forced to learn an "optimal" latent space: this space must be compact enough (satisfying the SVDD goal) and retain enough valid information to complete the reconstruction task (satisfying the VAE goal). This fundamentally avoids the "hypersphere collapse".

During detection, the two scores can be combined: the distance score is used to calculate the distance from the potential representation z of the test sample to the center

doi:10.6342/NTU202503439

c, and the reconstruction score is used to calculate the difference between the test sample x and its reconstructed image x'.

In this way, by introducing the reconstruction task as a regularization method, it is ensured that the model can learn meaningful features, effectively prevent the collapse of the hypersphere, and significantly improve the stability of training and the effectiveness of the final model. At the same time, since the model can provide both distance-based anomaly scores and reconstruction-based anomaly scores, the two scores can complement each other. For example, some anomalies may be close to the center in the latent space, but difficult to reconstruct, and vice versa. Combining the two can improve the robustness of detection.

Secondly, our datasets have limitations, and our Image AUC performance is not particularly outstanding. Therefore, in the future, we will try to test on more universal but accurate anomaly detection datasets, such as MVTec AD [14] and VisA [15], to test the versatility and practicality of the algorithm. We will also try to improve the original datasets. First, we can manually check the datasets to improve the problem of Z-AOI false detection. Secondly, we can develop our own definition of glucose enzyme test strip anomalies to strictly regulate anomalies and reduce ambiguous situations. Then, for those test strips with anomalies but ignored for various reasons, there is an additional algorithm

doi:10.6342/NTU202503439

to define them and generate masks in the process of generating datasets. Finally, we can improve the hardware design of the inspection machine (as shown in Figure 6-1). The original machine is placed on the conveyor plane to scan the test strip, but in fact, in addition to looking straight, the test strip needs to be tilted at a certain angle to reflect light to see anomalies, and it also needs to face the light source to see if the silver wire is broken from the back. Because it is a silver wire covered with black carbon powder, if the silver wire is not broken, it can still be used. Therefore, the improved AOI machine should stand the test strip upright, illuminate the front, and use three cameras for front and side view, and back view.

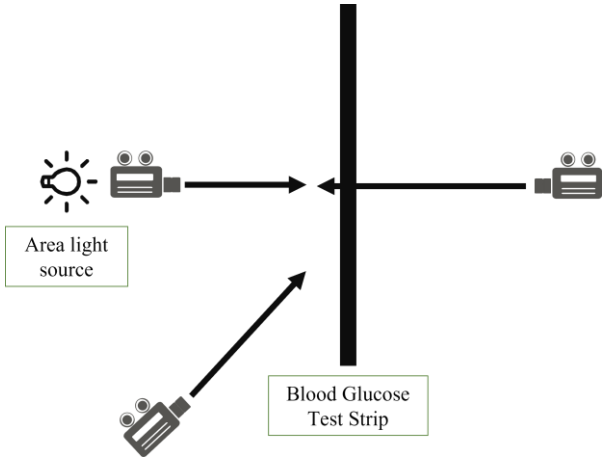


Figure 6-1: Future machine design.

doi:10.6342/NTU202503439

References

[1] A. Mousakhan, T. Brox, and J. Tayyub “Anomaly Detection with Conditioned Denoising Diffusion Models,” https://arxiv.org/abs/2305.15956, 2023.

[2] G. E. Hinton and R. R. Salakhutdinov, “Reducing the Dimensionality of Data with Neural Networks,” Science, Vol. 313, No. 5786, pp. 504-507, DOI:10.1126/science.1127647, 2006.

[3] D. P. Kingma and M. Welling, “Auto-Encoding Variational Bayes,” https://arxiv.org/abs/1312.6114, 2013.

[4] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. “Generative Adversarial Nets.” Neural Information Processing Systems, https://arxiv.org/abs/1406.2661, 2014.

[5] L. Ruff., R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller and M. Kloft. “Deep One-Class Classification.” Proceedings of the 35th International Conference on Machine Learning, PMLR 80:4393-4402, https://proceedings.mlr.press/v80/ruff18a.html, 2018.

[6] D.M. Tax, R.P. Duin. “Support Vector Data Description.” Machine Learning 54, 45–66, https://doi.org/10.1023/B:MACH.0000008084.60811.49, 2004.

[7] Huang, W.; Li, Y.; Xu, Z.; Yao, X.; Wan, R. “Improved Deep Support

doi:10.6342/NTU202503439

Vector Data Description Model Using Feature Patching for Industrial Anomaly Detection.” Sensors 2025, 25, 67. https://doi.org/10.3390/s25010067.

[8] Deng X, Zhang Z. “Nonlinear Chemical Process Fault Diagnosis Using Ensemble Deep Support Vector Data Description.” Sensors. 2020; 20(16):4599. https://doi.org/10.3390/s20164599.

[9] Schlegl, T., Seeböck, P., Waldstein, S. M., Langs, G., & Schmidt-Erfurth, U. (2019). “f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks.” Medical image analysis, 54, 30-44.

[10] Batzner, K., Heckler, L., & König, R. (2024). “Efficientad: Accurate visual anomaly detection at millisecond-level latencies.” In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 128-138).

[11] Zhou, Y., Liang, X., Zhang, W., Zhang, L., & Song, X. (2021). “VAE-based deep SVDD for anomaly detection.” Neurocomputing, 453, 131-140.

[12] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, “Uninformed Students: Student-Teacher Anomaly Detection with Discriminative Latent Embeddings,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, pp. 4183–4192, doi: 10.1109/CVPR42600.2020.00424, 2020

doi:10.6342/NTU202503439

[13] M. Teodorczyk, M. Cardosi, and S. Setford, “Hematocrit Compensation in Electrochemical Blood Glucose Monitoring Systems,” *Journal of Diabetes Science and Technology,* vol. 6, no. 3, pp. 648-655, doi: 10.1177/193229681200600320, 2012.

[14] Bergmann, Paul, et al. "MVTec AD--A comprehensive real-world dataset for unsupervised anomaly detection." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019.

[15] Zou, Y., Jeong, J., Pemula, L., Zhang, D., & Dabeer, O. (2022, October). Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In European Conference on Computer Vision (pp. 392-408). Cham: Springer Nature Switzerland.

doi:10.6342/NTU202503439